



# **BIOMETRIC AUTHENTICATION AND AUTHORISATION INFRASTRUCTURES**

---

Dissertation zur Erlangung des Grades eines Doktors der Wirtschaftswissenschaften  
eingereicht an der Wirtschaftswissenschaftlichen Fakultät der Universität Regensburg

vorgelegt von

Dipl.-Wirt.Inf. Matthias Olden

Berichterstatter  
Prof. Dr. Dieter Bartmann  
Prof. Dr. Günther Pernul

Regensburg, den 21. Oktober 2008

## PREFACE

Nowadays, replacing traditional authentication methods with authentication and authorization infrastructures (AAIs) comes down to trading several passwords for one “master password”, which allows users to access all services in a federation. Having only one password may be comfortable for the user, but it also raises the interest of potential impostors, who may try to overcome the weak security that a single password provides. A solution to this issue would be a more-factor AAI, combining the password with a biometric method of authentication that can work on the internet. The model presented in this work is based on typing behaviour biometrics, which can recognize a user by the way he types (Bartmann 2007). This biometric method uses the keyboard as a sensor and is a pure software solution that can function in a web browser.

Due to the fact that biometrics do not require any knowledge-based features (like passwords), biometric AAI based on typing behaviour are comfortable for the user. Also, no special devices (like tokens) are necessary for the authentication. Additionally, biometric AAI provide high protection against attacks by uniquely assigning a username to a certain person. These advantages make biometric AAI interesting for practical use.

As common AAI were not especially designed to be used with biometrics (Schläger 2008), their architectures do not foresee specific biometric issues like the process of enrolment on different servers, template aging and synchronisation of biometric data (e.g. for the purpose of recognizing replay attacks). They also do not include methods of delivering information about the quality of biometric data upon the login process. A part of this research will concentrate itself upon the problems of biometrics in combination with AAI, which will be studied both at the level of the typing behaviour biometric as well as at the level of AAI. For this, different AAI architectures will be investigated in order to see whether they permit the use of biometrics as authentication technology and to research the necessary changes in their architectures in order to provide a reference model for a biometric AAI.

## LOGIC FLOW DIAGRAM

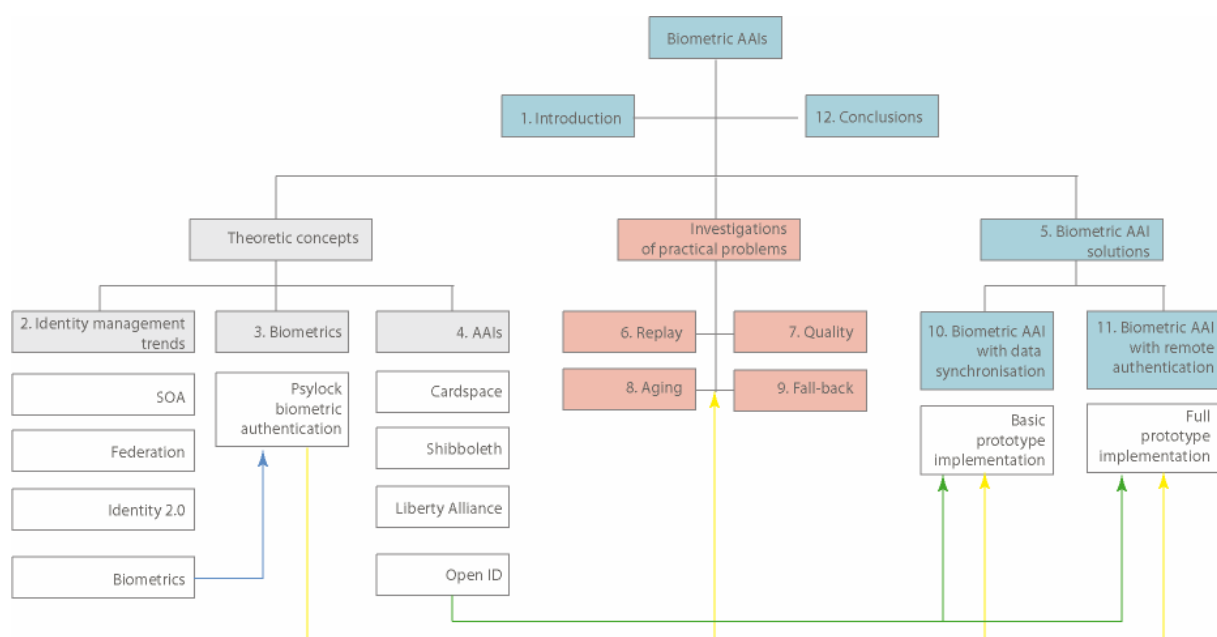
This work is divided in three parts:

I. Theoretical concepts: In this first part, different concepts concerning identity management, biometric authentication and AAIs are investigated at a theoretic level. The various trends in identity management systems show the necessity of increasing security by the use of biometrics. This makes it important to understand the particularities of biometric systems, which will be done on the example of typing cadence. Furthermore, criteria for the choice of an AAI appropriate for biometric integration will be elaborated.

II. Investigation of practical issues: This part of the work is an in-depth view on the problems of biometric authentication. Several issues like replay attacks, quality and aging of biometric data are researched by means of examples and experiments taken from typing behaviour biometrics. Another investigation topic is the conception of fall-back mechanisms for more-factor authentication.

III. Biometric AAI solutions: This part includes the development of use-cases and real prototypes of biometric AAIs. For this purpose, two possible solutions are provided for different system architectures.

A logic flow diagram of this work is presented here:



# CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
<b>1.1</b>	<b>Problematic.....</b>	<b>1</b>
<b>1.2</b>	<b>Purpose of this work.....</b>	<b>3</b>
1.2.1	Particularities of the use of AAIs together with biometrics.....	3
1.2.2	Conception of an architectural model for biometric authentication services .....	3
<b>1.3</b>	<b>Research questions.....</b>	<b>3</b>
1.3.1	Architectural aspects: aging process of biometric data.....	4
1.3.2	Security aspects: replay attacks .....	4
1.3.3	Quality aspects: quality of biometric features.....	4
1.3.4	Consequences for architectures: reference models .....	5
1.3.5	Prototype implementation of a biometric AAI on the basis of typing behaviour .....	5
<b>2</b>	<b>IDENTITY MANAGEMENT .....</b>	<b>6</b>
<b>2.1</b>	<b>Reasons for using identity management .....</b>	<b>6</b>
<b>2.2</b>	<b>Definition of terms .....</b>	<b>7</b>
2.2.1	Identity .....	7
2.2.2	Partial identity .....	7
<b>2.3</b>	<b>Identity management.....</b>	<b>8</b>
<b>2.4</b>	<b>Functionality and components of an IDM system .....</b>	<b>8</b>
2.4.1	The level of personal data.....	9
2.4.2	The level of resources .....	9
2.4.3	The level of authentication .....	9
2.4.4	The level of authorisation .....	10
<b>2.5</b>	<b>Trends in the field of IDM .....</b>	<b>11</b>
2.5.1	The number of IDM providers will increase.....	11
2.5.2	Companies will use federated identity management .....	12
2.5.3	Privacy and data protection will be gaining importance.....	12
2.5.4	Identity 2.0 will be the base of future IDM systems.....	13
2.5.5	Biometrics will contribute to increase the security of IDM systems.....	15
<b>2.6</b>	<b>Evaluation.....</b>	<b>16</b>
<b>3</b>	<b>BIOMETRICS .....</b>	<b>17</b>
<b>3.1</b>	<b>Motivation.....</b>	<b>17</b>
<b>3.2</b>	<b>Terminology.....</b>	<b>18</b>
<b>3.3</b>	<b>Typing cadence as a biometric method .....</b>	<b>22</b>
3.3.1	Classification of typing cadence biometrics.....	23
3.3.2	Criteria for biometric features .....	24
3.3.3	Criteria for biometric methods .....	25
3.3.4	Particularities of typing cadence.....	26
3.3.5	Operational areas .....	26
3.3.6	Typing cadence by Psylock .....	27

<b>4</b>	<b>AUTHENTICATION AND AUTHORISATION INFRASTRUCTURES .....</b>	<b>29</b>
4.1	Definition and role of AAI .....	29
4.2	Requirements analysis.....	30
4.3	Basic concepts of AAI systems .....	31
4.3.1	AAI components .....	31
4.3.2	Ticket systems.....	32
4.3.3	Circle of Trust .....	32
4.3.4	Central Single Sign On server .....	33
4.4	Considered AAI systems .....	34
4.4.1	Central Authentication Service (CAS).....	34
4.4.2	Shibboleth .....	35
4.4.3	Liberty Alliance .....	37
4.4.4	Windows CardSpace.....	37
4.4.5	Sxip .....	39
4.4.6	OpenID.....	39
4.4.6.1	Concepts of OpenID .....	39
4.4.6.2	How OpenID works.....	40
4.4.6.3	New features of OpenID 2.0.....	42
4.4.6.3.1	Better extensions support.....	42
4.4.6.3.2	Large requests and replies .....	42
4.4.6.3.3	Directed Identity .....	43
4.4.6.3.4	Provider Authentication Policy Extension (PAPE) .....	44
4.4.6.4	OpenID as implementation platform.....	44
<b>5</b>	<b>BIOMETRIC AAIS .....</b>	<b>46</b>
5.1	Authentication methods in AAIs.....	46
5.2	Architectural models .....	48
5.3	Problems of biometrics that influence the biometric AAIs.....	49
5.3.1	Replay attack as a problem for AAI systems .....	50
5.3.2	Quality of biometric data as a problem for biometric AAIs .....	51
5.3.3	Aging of biometric data as a problem for biometric AAIs .....	52
5.4	Conclusion .....	53
<b>6</b>	<b>REPLAY ATTACKS IN BIOMETRIC SYSTEMS BASED ON TYPING BEHAVIOUR.....</b>	<b>55</b>
6.1	Security problems in IT-systems.....	55
6.2	Security problems of biometric systems.....	56
6.3	Replay attacks .....	57
6.3.1	Protection against replay attacks .....	58
6.4	Key logging .....	59
6.4.1	Susceptibility for replay attacks .....	60
6.5	Replay Algorithm.....	62
6.5.1	Core of the checkReplay function .....	65
6.5.2	Test environment .....	68
6.5.3	Test phases .....	69

<b>6.6</b>	<b>Extending the test procedure.....</b>	<b>75</b>
6.6.1	Requirements to the new test scenario .....	77
6.6.2	Extending the generation process of the replay sample.....	77
6.6.3	Including the match rate of the biometric system as additional feature .....	79
6.6.4	Connecting the replay algorithm to the biometric API.....	80
6.6.5	New test results .....	81
<b>6.7</b>	<b>Conclusion .....</b>	<b>83</b>
<b>7</b>	<b>QUALITY TESTS FOR BIOMETRIC SYSTEMS.....</b>	<b>84</b>
<b>7.1</b>	<b>Quality problems of biometric systems .....</b>	<b>84</b>
<b>7.2</b>	<b>Recording key events with typing behaviour biometrics .....</b>	<b>86</b>
<b>7.3</b>	<b>Software problems .....</b>	<b>87</b>
7.3.1	Raster tests .....	88
7.3.2	Key code recognition tests.....	90
7.3.2.1	Key code recognition in Flash .....	90
7.3.2.2	Key code recognition in JavaScript.....	91
7.3.3	Speed-delay tests.....	92
7.3.3.1	Speed-delay tests in Flash.....	92
7.3.3.2	Speed-delay test in JavaScript .....	93
7.3.4	Foreign language compatibility.....	93
7.3.5	Enrolment – authentication analysis.....	95
<b>7.4</b>	<b>Hardware problems (different keyboards).....</b>	<b>97</b>
7.4.1	Test procedure.....	98
7.4.2	Expected results .....	99
7.4.3	Test results .....	102
7.4.4	Conclusion .....	107
<b>8</b>	<b>AGING OF BIOMETRIC FEATURES .....</b>	<b>108</b>
<b>8.1</b>	<b>Aging of the reference template .....</b>	<b>108</b>
<b>8.2</b>	<b>Experimental setup.....</b>	<b>109</b>
<b>8.3</b>	<b>Feature extraction.....</b>	<b>112</b>
<b>8.4</b>	<b>Time dependent features.....</b>	<b>113</b>
8.4.1	N-segment duration.....	113
8.4.1.1	Calculation .....	113
8.4.1.2	Expectations.....	114
8.4.1.3	Analysis.....	115
8.4.2	Speed.....	116
8.4.2.1	Calculation .....	116
8.4.2.2	Expectation.....	117
8.4.2.3	Analysis.....	117
8.4.3	Outliers.....	118
8.4.3.1	Calculations.....	118
8.4.3.2	Expectations.....	119
8.4.3.3	Analysis.....	120
8.4.4	Crossovers.....	121
8.4.4.1	Calculation .....	121
8.4.4.2	Expectations.....	122
8.4.4.3	Analysis.....	123
<b>8.5</b>	<b>Time independent features .....</b>	<b>124</b>
8.5.1	Typing mistakes and correction behaviour .....	124

8.5.1.1	Calculation .....	124
8.5.1.2	Expectations .....	125
8.5.1.3	Analysis .....	126
<b>8.6</b>	<b>Conclusions .....</b>	<b>126</b>
<b>9</b>	<b>DESIGNING A FALL-BACK SOLUTION FOR A MULTI-FACTOR AUTHENTICATION USING BIOMETRICS.....</b>	<b>128</b>
<b>9.1</b>	<b>Multiple factor authentication.....</b>	<b>128</b>
<b>9.2</b>	<b>Key management .....</b>	<b>129</b>
<b>9.3</b>	<b>Fall-back mechanism.....</b>	<b>131</b>
<b>9.4</b>	<b>Fall-back problems .....</b>	<b>133</b>
<b>9.5</b>	<b>Conclusion .....</b>	<b>134</b>
<b>10</b>	<b>BIOMETRIC AAIS WITH SYNCHRONISED DATA .....</b>	<b>135</b>
<b>10.1</b>	<b>Introduction.....</b>	<b>135</b>
10.1.1	Combination of biometric methods with AAIs.....	135
<b>10.2</b>	<b>Problems and requirements of a Circle of Trust .....</b>	<b>136</b>
10.2.1	Single Sign On .....	136
10.2.2	Attribute management .....	136
10.2.3	Assignment of user names.....	137
10.2.3.1	User names valid for the entire Circle of Trust.....	137
10.2.3.2	Individual user names for every application .....	137
10.2.3.2.1	Use of a mapping table .....	137
10.2.3.2.2	Dynamic assignment of accounts by means of biometrics.....	139
10.2.4	Mirroring of biometric accounts on the example of Psylock.....	140
10.2.4.1	Psylock data to transfer.....	140
10.2.4.2	Necessary actuality due to replay attacks.....	142
10.2.4.3	Synchronisation failures .....	142
<b>10.3</b>	<b>Synchronisation on the database level.....</b>	<b>143</b>
<b>10.4</b>	<b>OpenID Attribute Exchange Extension .....</b>	<b>144</b>
<b>10.5</b>	<b>Scenarios for a circle of trust with OpenID .....</b>	<b>148</b>
10.5.1	1 <sup>st</sup> configuration: one identity provider and more consumers .....	148
10.5.1.1	Enrolment workflow .....	150
10.5.1.2	Biometric login at the IdP.....	153
10.5.1.3	Biometric login at the consumers .....	153
10.5.2	2 <sup>nd</sup> configuration: a server is used as consumer or as IdP.....	155
10.5.3	3 <sup>rd</sup> configuration: a user has several IdPs that have also consumer functionality .....	158
10.5.3.1	Enrolment workflow .....	159
10.5.3.2	Authentication workflow.....	159
10.5.4	4 <sup>th</sup> configuration: a user can have more IdPs for a consumer .....	160
10.5.5	5 <sup>th</sup> configuration: an application supports all possible configurations at the same time.....	161
<b>10.6</b>	<b>Conclusion .....</b>	<b>163</b>
<b>11</b>	<b>BIOMETRIC AAIS WITH REMOTE AUTHENTICATION.....</b>	<b>164</b>
<b>11.1</b>	<b>Introduction.....</b>	<b>164</b>

<b>11.2</b>	<b>Possible solutions.....</b>	<b>166</b>
11.2.1	Changes in the discovery process.....	167
11.2.2	Changes in the assertion process.....	167
11.2.3	Choosing the right solution.....	167
<b>11.3</b>	<b>The CoT-Logic .....</b>	<b>169</b>
11.3.1	Ways of using the CoT-Logic .....	172
11.3.1.1	CoT-Logic in standalone mode .....	172
11.3.1.2	CoT-Logic in full server mode .....	173
11.3.2	Division between the CoT-Logic and the IdP.....	174
11.3.3	Data storing of the CoT – Logic instances.....	175
11.3.4	Communication of CoT-Logic instances .....	177
11.3.4.1	Secure communication.....	177
11.3.4.2	Consumer management .....	178
11.3.4.3	CoT-Logic instance management.....	178
<b>11.4</b>	<b>Remote Authentication.....</b>	<b>179</b>
11.4.1	Definition .....	179
11.4.2	Functionality of remote authentication.....	181
11.4.2.1	Integration .....	181
11.4.2.2	Checking the foreign IdP.....	181
11.4.2.3	Representation of assertion relationships .....	182
11.4.3	Consumer mode .....	182
11.4.3.1	Mapping the authentication request of the consumer to the authentication response of the home IdP .....	183
11.4.4	Mapper .....	184
11.4.5	Prototype demo .....	185
<b>11.5</b>	<b>Advantages of using biometrics for the participating parties .....</b>	<b>187</b>
11.5.1	User .....	187
11.5.2	Identity provider.....	187
11.5.3	Service provider (consumer) .....	188
<b>11.6</b>	<b>Conclusion .....</b>	<b>188</b>
<b>12</b>	<b>CONCLUSIONS AND FUTURE WORK .....</b>	<b>190</b>
<b>12.1</b>	<b>Conclusions.....</b>	<b>190</b>
<b>12.2</b>	<b>Future work.....</b>	<b>192</b>

## LIST OF FIGURES

<i>Number</i>	<i>Page</i>
Fig. 2-1 Partial identity according to (Jendricke 2003) .....	8
Fig. 2-2 Increase of digital identities. On the basis of (Lukawiecki 2006) .....	13
Fig. 2-3 Identity 1.0 is site centric. On the basis of (Hardt 2005) .....	14
Fig. 2-4 Identity 1.0, on the basis of (Hardt 2005) .....	14
Fig. 2-5 Identity 2.0, on the basis of (Hardt 2005) .....	15
Fig. 3-1 Typical internal enrolment process (Bromba 2008) .....	19
Fig. 3-2 Functionality of biometrics .....	20
Fig. 3-3 FAR/FRR curve .....	21
Fig. 3-4 Identification and enrolment process (Pike 2008; Bromba 2008) .....	22
Fig. 3-5 Psylock in comparison to other biometrics (Centre for Mathematics, 2002) .....	28
Fig. 4-1 Single Sign On .....	34
Fig. 4-2 Shibboleth architecture (Swiss Education 2007) .....	36
Fig. 4-3 CardSpace functionality (CardSpace 2008) .....	38
Fig. 4-4 How OpenID works .....	40
Fig. 5-1 Biometric authentication in a circle of trust requires changes in both IdP and biometric component .....	47
Fig. 5-2 Biometric AAI architectures .....	48
Fig. 5-3 Replay in biometric AAIs .....	50
Fig. 5-4 Quality problems in biometric AAIs .....	52
Fig. 5-5 Aging in biometric AAIs .....	53
Fig. 6-1 Replay attack scenarios (Ratha 2001) .....	61
Fig. 6-2 Array generated from a sample .....	67
Fig. 6-3 Logic flow of the replay algorithm .....	68
Fig. 6-4 Original vs. 5 typing samples from the same users .....	70
Fig. 6-5 Original vs. 5 replay samples .....	70
Fig. 6-6 FAR for “type 1” replay .....	72
Fig. 6-7 FAR for “type 2” replay .....	73
Fig. 6-8 FAR for “type 3” replay .....	73
Fig. 6-9 Replay FRR for original samples (“type 0”) .....	74
Fig. 6-10 Replay FAR and FRR curves .....	75
Fig. 6-11 FAR curve for “type 2” replay – trend .....	76
Fig. 6-12 Connecting the replay algorithm to the biometric API .....	80
Fig. 6-13 Replay and biometric match score for original samples .....	82
Fig. 6-14 Replay and biometric match score for replay samples .....	83
Fig. 7-1 Resolution tests under Windows .....	88
Fig. 7-2 Resolution tests under LINUX .....	89
Fig. 7-3 Resolution tests under MAC .....	89
Fig. 7-4 Speed-delay in Flash for Mozilla, IE and Opera .....	93
Fig. 7-5 Match scores reached by different browsers while authenticating to a biometric profile created with Opera 8 .....	95
Fig. 7-6 Match scores reached by different browsers while authenticating to a biometric profile created with Netscape .....	96
Fig. 7-7 Matching scores reached by different browsers while authenticating to a biometric profile created with Internet Explorer .....	96
Fig. 7-8 EER dependence of the number of enrolment samples (Achatz 2006) .....	99
Fig. 7-9 Match scores by keyboard change without adaption .....	100
Fig. 7-10 Adaption of the template leads to higher match scores .....	101

Fig. 7-11 Authentication to a multi-keyboard enrolment template without adaption .....	101
Fig. 7-12 Authentication to a multi-keyboard enrolment template without adaption .....	102
Fig. 7-13 Quality of the typing samples without the adaption .....	102
Fig. 7-14 Different keyboards without adaption.....	103
Fig. 7-15 Template adaption .....	104
Fig. 7-16 Template adaption with multiple keyboards.....	104
Fig. 7-17 Mixed profile while attempting to log in with all keyboards .....	106
Fig. 7-18 FAR and FRR curves of the mixed profile .....	107
Fig. 8-1 Experimental setup to determine the aging process of typing behaviour biometric ..	110
Fig. 8-2 The feature processing chain (Bakdi 2007) .....	113
Fig. 8-3 Expected development of the n-segment duration .....	115
Fig. 8-4 Actual development of n-segment duration .....	115
Fig. 8-5 Expected development of speed.....	117
Fig. 8-6 Actual development of speed .....	117
Fig. 8-7 Expected development of outliers .....	120
Fig. 8-8 Actual development of outliers .....	120
Fig. 8-9 Expected development of crossovers .....	123
Fig. 8-10 Actual development of crossovers .....	123
Fig. 8-11 Expected development of typing mistakes .....	125
Fig. 8-12 Actual development of typing mistakes.....	126
Fig. 9-1 Key management – Generation and storage of keys.....	130
Fig. 9-2 Fall-back mechanism in case of a forgotten password .....	133
Fig. 10-1 Use of a central mapping table .....	138
Fig. 10-2 Mapping table stored by each IdP in the circle of trust.....	139
Fig. 10-3 Simplified biometric database structure.....	140
Fig. 10-4 Central repository.....	143
Fig. 10-5 The decentralized version.....	144
Fig. 10-6 First configuration .....	149
Fig. 10-7 Enrolment workflow .....	150
Fig. 10-8 Second use-case.....	151
Fig. 10-9 Biometric login at the IdP .....	153
Fig. 10-10 Biometric login at the consumers .....	154
Fig. 10-11 The second configuration .....	155
Fig. 10-12 Original database structure of an identity provider.....	156
Fig. 10-13 Original database structure of a consumer.....	157
Fig. 10-14 Combined database model.....	158
Fig. 10-15 The third configuration .....	159
Fig. 10-16 The fourth configuration.....	160
Fig. 10-17 The fifth configuration .....	162
Fig. 10-18 Final database model.....	162
Fig. 11-1 Circle of trust with biometric AAIs.....	165
Fig. 11-2 Ranking process of possible solutions .....	169
Fig. 11-3 The CoT-Logic.....	170
Fig. 11-4 Logic flow of the first CoT-Logic variant .....	172
Fig. 11-5 Logic flow of the first CoT-Logic variant .....	173
Fig. 11-6 Division between the CoT-Logic and the IdP functionality .....	174
Fig. 11-7 Data storage of the CoT-Logic instance.....	177
Fig. 11-8 Adding a new CoT-Logic instance to the circle.....	179
Fig. 11-9 Problems without remote authentication.....	180
Fig. 11-10 Logic flow of the prototype.....	185

## LIST OF TABLES

<i>Number</i>	<i>Page</i>
Table 3-1 Classification of biometrics after (Bromba 2008) .....	23
Table 3-2 Comparison of various biometric technologies, modified from (Jain 2004).....	24
Table 6-1 Risks of biometric systems and countermeasures (ISACA Group 2008) .....	57
Table 6-2 Replay attack attempts .....	62
Table 6-3 Replay and biometric match score for original samples .....	82
Table 6-4 Replay and biometric match score for replay samples.....	82
Table 7-1 Tests with browser-OS combinations.....	88
Table 7-2 Key code recognition in Flash.....	90
Table 7-3 Key code recognition in JavaScript .....	91
Table 7-4 Results of the enrolment – authentication analysis .....	96
Table 9-1 Biometric enrolment with fall-back option.....	131
Table 9-2 Biometric authentication with fall-back option.....	132
Table 10-1 Biometric database.....	141
Table 11-1 Criteria for designing a circle of trust with OpenID .....	168

## AKNOWLEDGEMENT

I would like to thank Prof. Dr. Dieter Bartmann for the excellent mentoring, motivation, enthusiasm and support that he offered during the making of this dissertation and for the strong belief he had in me.

I am also grateful to Prof. Dr. Günther Pernul for the solid ideas and suggestions he gave during the making of this work. His broad experience in the field of AAI's helped me to overcome the complexity of the topic and to concentrate upon the relevant facts.

Last but not least, I show gratitude to all the students that have helped me by working together with me on different projects. Without these extraordinary people it would not have been possible to make this work.

## ACRONYMS

### Abbreviation

AAI = Authentication and Authorisation Infrastructure

API = Application Programming Interface

AX = Attribute Exchange Extension

CoT = Circle of Trust

IdM = Identity Management

IdP = Identity Provider

IP = Internet Protocol

JVM = Java Virtual Machine

OS = Operating System

PAPE = Provider Authentication Policy Extension

PKI = Public Key Infrastructure

RP = Relaying Party

SAML = Security Assertion Mark-up Language

SOA = Service Oriented Architecture

SP = Service Provider

sREG = Simple Registration Extension

SSO = Single Sign On

URI = Uniform Resource Identifier

URL = Uniform Resource Locator

XRDS = eXtensible Resource Descriptor Sequence

# *Chapter 1*

## 1 INTRODUCTION

---

This chapter gives an overview of the current situation, where the high number of providers makes it impossible for one user to manage so many passwords. AAI's can be a solution to this problem, but only if their authentication is improved. The suggested proposition is the use of typing behaviour biometrics as an authentication method for an AAI. Possible biometric specific problems have to be considered.

---

### 1.1 Problematic

Today, with the rapid growth of internet and the introduction of Web 2.0, the rules the internet is based on are changing. The old model where the providers and the consumers of web services were two separate entities is being replaced by the new possibilities of web technology, which allow anybody who is online to be both provider and consumer. These new opportunities make the internet attractive to an increased number of companies providing services to a large number of users.

This new trend has to be put in correlation with the different security policies that companies (web providers) follow and with the influence that these policies have upon users. Seen from the side of the web providers, good security policies establish *who* is allowed to use a system and *in which circumstances* they are allowed to use it (Stein 2003). On the side of the users, the different security policies are reflected in an increased number of credentials, mostly in the form of a username / password combination. This large number of passwords leads to users tending to choose simple combinations or to use the same password for more services. Against this practice, some web service providers protect themselves by checking passwords against common dictionary entries or by implementing special rules which require that passwords should be long, with small and capital letters, numbers and special characters. With these restrictions, passwords are often forgotten or written down, which brings other risks and security leaks.

The immediate consequence of this development is that the username / password combination has reached its limits and other ways of authentication must be researched. One of them is the Single Sign On, that is very similar to a password manager. Its advantage is that it grants access to all web services by means of a one time authentication. Despite of this comfortable feature, the Single Sign On does not add security to the system. Another disadvantage is the necessary synchronisation of all security policies of the web services managed, which implies that the SSO has to be able, for example, to change all passwords before expiration date according to the respective security policies. In the classic web authentication, every web provider is responsible for the credentials of its users, while the Single Sign On (for example Microsoft Passport) stores this sensitive data on a central server, thus making it a target for different types of attacks (Korman 2000). These considerations prove that the Single Sign On cannot comply with the expectations of the future internet world.

A solution to the problems mentioned above is offered by authentication and authorisation infrastructures (called AAIs from now on), which are combinations of services and methods allowing customers of different web services the access to protected contents stored on different servers. In this case, the authentication does not take place on every server, nor in some central place, but on the server of one single company, which later submits the authorisation to another web service requesting it.

Although the AAIs represent the successor of Single Sign On technology, their principles of functioning are not yet clearly defined and many questions are still to be answered (Schläger 2007).

So far, there are implementations of different AAIs based on password technology. However, these have the disadvantages that come with the knowledge factor of password. In the case of the AAIs, a user is granted access to all of his accounts with one authentication (thus having one single internet identity); it is indispensable that no other user is able to falsely authenticate as someone else. This request makes password and token based authentications incompatible with future AAIs. The only authentication method which can provide protection against the passing on of user credentials is biometrics.

In use with AAIs, authentication methods based on biometrics present several advantages, like the possibility to uniquely identify a user, the impossibility of assuming someone else's identity and the fact that they do not require credentials to be memorized (like passwords) or carried along (like tokens). These advantages make biometric AAIs a solution that answers the demands of the future web community.

## **1.2 Purpose of this work**

This work concentrates upon the research of biometric authentication technologies in combination with AAIs. This will be followed both at the level of architectural concepts as well as at the level of practical implementation. This results in two main research topics:

### **1.2.1 Particularities of the use of AAIs together with biometrics**

While biometric methods provide an authentication technology which is already used in practice, their implementation within an AAI raises a set of special questions. These questions are general ones, occur for every biometric method and can be roughly classified in:

- Architectural problems (e.g adaption, profile distribution, frequency of use, template aging);
- Security problems (feature theft, replay attacks);
- Quality problems (quality of enrolment process, quality of feature extractors).

The purpose of this work is to investigate these problems and to provide solutions to them.

### **1.2.2 Conception of an architectural model for biometric authentication services**

So far, there is no solution for implementing biometrics within AAIs. Therefore, it is necessary to investigate the current architecture models for AAIs in respect to their compliance with biometric standards. If necessary, the architectures of AAIs have to be modelled especially in order to work with biometrics.

Based on standard AAI architectures and using the research results, a reference model for biometric AAIs has been developed. This model has been implemented as a prototype. The biometric method used for this prototype is the typing behaviour described in (Bartmann 2000) and (Bartmann 2004).

## **1.3 Research questions**

This work will present several biometric problems that occur in the context of AAIs, such as:

### **1.3.1 Architectural aspects: aging process of biometric data**

One important characteristic of biometrics is the fact that biometric data changes through time, independent on the type of biometrics. It is therefore necessary to examine the role of aging of biometric data within an AAI, whose architecture requires many biometric profiles of the same user on different servers that are not all regularly actualized. Due to the fact that this problem occurs for every type of biometrics in a similar way, solutions for this problem are provided in a general manner for all biometrics.

### **1.3.2 Security aspects: replay attacks**

Important attention is given to the problem of theft of biometric data and to the possibility of preventing it by means of algorithms that recognize replay attacks. Due to the fact that every biometric method has its own particular way of treating such attacks (and a different degree of vulnerability against replay attacks), this PhD work discusses only the possibility of replaying data for typing cadence biometrics. For this biometric method, there is currently no efficient protection against replay attacks.

The second challenge is the real time replay checking of biometric data stored on different servers. The method presented in this work can be applied for all types of biometrics.

### **1.3.3 Quality aspects: quality of biometric features**

The quality that biometrics deliver depends very much on the way in which the user enrolls and on the type of sensor he is using. This quality problem is of high importance for biometric AAIs, due to the fact that they have to support all combinations of software solutions as well as hardware sensors.

This PhD work researches a method apt to check the quality of biometric data and to deliver useful information about a possible increase in quality. This method uses general functions of biometrics described in biometric interfaces like (BioAPI 2008), in order to determine in real time values like FAR, FRR and EER, while making use of biometric data located on various servers.

For the case of typing behaviour biometrics, it was also researched which kind of other quality indicators can be determined.

#### **1.3.4 Consequences for architectures: reference models**

The current common AAI's are not especially designed to be used with biometrics. Therefore, their architectures do not foresee the process of enrolment on different servers, the changes which may appear in biometric data over a period of time, the synchronisation of biometric data for the purpose of checking a replay attack or methods of delivering information about the quality of biometric data upon the login process. At the same time, interchanging biometric data (which has a much higher value than a normal password, due to the fact that it cannot be replaced in case of corruption) can raise significant security questions. Therefore, a solution was researched at the level of the architecture of the AAI. A list of biometric attributes was generated and it was decided which one of them can be passed forward at the request of another server and which ones have to be kept locally for security reasons.

For this all known forms of AAI's were investigated in respect to their ability to permit the use of biometrics in their architectures. Possible changes were documented in order to provide a reference model for a biometric AAI.

#### **1.3.5 Prototype implementation of a biometric AAI on the basis of typing behaviour**

While researching the architecture and specific problem of biometric AAI's, new knowledge and information was gathered. It was relevant for this new knowledge to be implemented in the form of a prototype of biometric AAI based on the reference model elaborated. Due to the fact that all specific biometric problems were treated for the case of typing behaviour biometrics, this biometric method was implemented in an AAI. The advantages that typing behaviour provides lie in the fact that this biometric method does not require special sensors and therefore can easily be implemented as a replacement or enhancement for password protected AAI's.

## *Chapter 2*

### **2 IDENTITY MANAGEMENT**

---

Identity management is currently subject to a complete process of change, therefore the major trends in this field must be investigated. This chapter determines whether biometric methods can be seen as a possible future solution for identity management together with other major trends like SOA, federation, privacy or Identity 2.0.

---

#### **2.1 Reasons for using identity management**

In the field of IT technology, identity management has been playing an important role for many years. More and more executives recognise the importance of identity management and introduce such systems in their companies. According to a survey conducted by the Deron GmbH from Stuttgart in a cooperation project with the Fraunhofer Institute for Information and Communication Technology, 17% of the enterprises in the survey already use IDM systems, 7% are about to introduce them, and 38% plan a change to identity management. According to this survey, about two thirds of all enterprises favour an identity management solution. There are many reasons for this development which will be examined in the following. (Scherrbacher 2007)

The most important argument for identity management is the increase in corporate security, e.g. through system-wide user management. In an enterprise with several IT-Systems, the process of giving access authorisation is often inconsistent or antiquated. Frequently, users are granted more rights than they would need to do their respective task. When a user leaves the enterprise, his access authorisation has to be cancelled. What seems evident does not coincide with reality though, as shown by a study of CERT in cooperation with the United States Secret Service. According to this study, about 50% of all attacks on security in an enterprise are conducted by former employees (Geschonneck 2008). The reason for this is that the administrator has to cancel the rights of access of the employee in every existing system. In a company with several hundred systems, it is very difficult to fulfil this task, especially when it requires many administrators, each responsible for a small fraction of the systems. Because of the system-wide user management, IDM makes it possible

to cancel all user access rights at once and thus to prevent attacks from former employees with remaining rights. The same benefit appears when a user changes his department within the company, in which case the administrator can easily deactivate old rights and replace them with new ones. (Richter 2007; Parthier 2003; Mezler-Andelberg 2008)

Another effect of IDM is the increase in performance. With a system-wide user management, the administrator can add or delete user access rights with a single action. But in most cases, not even this is necessary, as the account of an employee is created in the human resource system upon his entry in the company; this account is accessible to all IT systems. At the same time, the rights of the new employee are defined by role systems, set up by the administrator. Rarely does the administrator have to intervene personally, e.g. when he consigns project-related access rights to an employee. As a rule, though, the user is assigned his role upon entering a department. When a department is dissolved, the administrator does not have to cancel the rights of every employee working there, but deactivates the access authorisation of the whole department. (Richter 2007)

## **2.2 Definition of terms**

For better understanding, it is necessary to name some terms which will gain importance during this work.

### **2.2.1 Identity**

Identity is a very broad term that can hardly be defined uniformly. In the context relevant for this work, identity defines a person as being unique through its personality and its relationship to its environment. We understand by the identity of a user the sum of definite, characteristic features that make him unique. This includes physical features such as the colour of hair or eyes, but also behavioural or biometric features such as DNA or retina, which clearly identify a person. (Meints 2006; Abels 2006)

### **2.2.2 Partial identity**

Every person has at least one identity, but he can gather others throughout his life. In this way, a person can take up different roles depending on how and whom he is interacting with. During the communication with different partners, a person assumes different roles and reveals different data about himself. The person changes his identity depending on whether he wants to remain anonymous or identify himself in parts or completely. His close friends will know personal details, the cashier in a shop will see his credit card, while the police will be shown his driver's licence. His

diary, however, will be known only to the person himself. Variations of these pieces of identity can reach up to a false appearance with falsified data for the purpose of deceit of the communication partner. When these pieces of identity are put together, they add up to the entire identity of an individual as shown in the following graph. (Jendricke 2003)

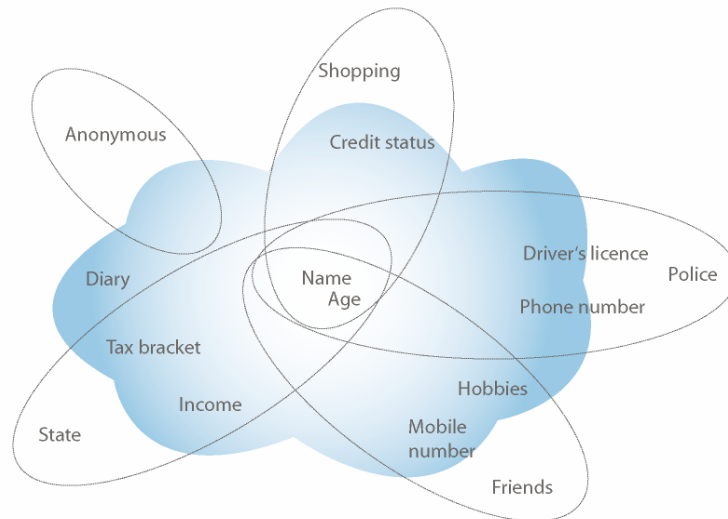


Fig. 2-1 Partial identity according to (Jendricke 2003)

## 2.3 Identity management

The aim and purpose of identity management is to make changes to identities only once and to validate them system-wide. The Burton Group, an independent market research company, defines identity management as „the set of business processes, and a supporting infrastructure for the creation, maintenance, and use of digital identities" (Burton 2008). It can be sub-divided in three classes: identity management in enterprises, between enterprises and identity management in the personal sector. The first, also called Identity and Access Management (IAM) or in-house IDM, is only related to one company and thus identity data can be used only within its limits. The identity management between enterprises, on the other hand, provides identity data independently of company limits. It is also known as FIM (Federated Identity Management). Lastly, there is person-related identity management, which is also called UCIM (User-Centric Identity Management). The user himself coordinates the identity management system and decides about the sort and the amount of information he provides about himself. In the following chapter the functionality of the IAM will be explained, leaving the FIM and UCIM for future research. (Mezler-Andelberg 2008)

## 2.4 Functionality and components of an IDM system

Generally, an IDM system can be represented in a model with four levels. These levels are:

- Personal data
- Resources
- Authentication
- Authorisation

These levels are connected and deliver or receive results to or from other levels by means of defined interfaces.

#### **2.4.1 The level of personal data**

The first level saves and manages personal data and provides the basis for all other levels. Personal data can be sub-divided into data of the persons that access the IT systems of an enterprise: employees, contractors and partners, customers. Employee data is usually recorded at the entry into the company by the HR department, contractor data is recorded in the purchasing department, and customer data in the sales department. (Mezler-Andelberg 2008)

#### **2.4.2 The level of resources**

The second level is the level of resources. Based on personal data from the first level, the level of resources creates user accounts, which then receive their access rights in the authorisation level. The resource level is divided in system areas and content areas. This division is necessary, as the authorisation level differentiates between access to functions and access to data. Resources are data saved in file systems or databases in the content area, as well as functions of programs in the system area. As only classified data requires special access rights, it can be divided by the objectives of availability and confidentiality. As of availability, data can be structured depending on its importance for the enterprise: the more important it is, the better accessible it should be.

#### **2.4.3 The level of authentication**

The third level, the level of authentication, has to state whether the user is the one he claims to be. If he identifies himself sufficiently, he is granted access to his user account, which lets him use applications and data. Identification can be effected in different ways, namely:

- knowledge-based methods
- token-based methods

- biometric methods
- hybrid methods.

The most frequent form of authentication is the knowledge-based method, usually expressed by passwords, but also pre-defined secret knowledge questions. Token-based methods assume the possession of a material object, e.g. a smart card. Biometric features, e.g. retina scan or fingerprinting, identify a person in the biometric methods. Depending on the required security level, not only one form of authentication is used, but a combination of these. This combination of two or more authentication techniques is called a hybrid method. A common example is the ATM, where a person uses his banking card and his PIN to withdraw. How secure a combination has to be depends on the risk and the effort. A risk would be the afore-mentioned data classification by confidentiality. The more important data or functions are, the higher security measures they require. The effort, on the other hand, increases with the number and the complexity of the authentication methods used.

Every method mentioned has advantages and disadvantages. Knowledge-based methods can be used without much effort and inexpensively. However, if a user has many passwords, he is prone to picking simple clues and/or writing them down as well as keeping them close at hand. If the user is careless, a password is an easy target for a potential attacker. A token does not need to be remembered, but it can be lost, stolen or passed on to an un-authorised person. While biometric methods show none of these disadvantages, for they can not be forgotten or lost, they need a costly and relatively elaborate realisation. They usually are very secure, but user acceptance is low. Beside this, persons and their biometric features change throughout their lives. (Mezler-Andelberg 2008)

#### **2.4.4 The level of authorisation**

The highest stands the level of authorisation, which mainly manages the rights for the users in an enterprise. The granting of rights serves the aforementioned objectives of confidentiality and integrity, as only authorised users should have access to data or applications. It is possible to grant specific rights manually to a single user, but this procedure is impracticable in a large enterprise with several thousand employees, and it is connected to a high administrative effort. In order to reduce this effort, users are united in groups so that the administrator can grant, modify or revoke rights to the entire group. This, however, carries problems, as an employee usually has rights to several systems, while groups are application specific. Thus, the rights for every system have to be managed separately. A solution to this problem is the role concept. By assuming a role, an employee can be member of many groups. In this way, roles can be understood as a wider group concept. A role is

not only a collection of users, but rather an intermediary between users and rights. This sort of access management is called RBAC (Role Based Access Control). RBAC was developed in 1992 by David Ferraiolo and Rick Kuhn from the American National Institute of Standards and Technology (NIST 2008). When a user changes the department, a change of rights can be conducted without much effort. The administrator deactivates a user's former role and assigns him another one. These simple operations are known as Core RBAC. However, as the rights management by means of roles is still complex, roles can be handed down. Enterprises mostly use a hierarchic structure, where the common employee stands lower than a member of management.

The so-called Constrained RBAC allows the definition of limitations or conditions. The result can be a separation of duties. A person can not take two roles that exclude each other, e.g. the roles of credit giver and credit taker - the owner of these roles could grant himself unlimited credit. In the level of authorisation, roles are created or deleted, given or withdrawn. The individual granting of rights that can not be covered by roles is found in this level as well. Generally, the distribution of rights follows the principle of least privilege, i.e. the user is granted only as many rights as he necessarily needs. (Mezler-Andelberg 2008; Todorov 2007; NIST 2008; Kowal 2004)

## **2.5 Trends in the field of IDM**

The development of IDM systems allows an estimation of a couple major trends, which will play an important role in the future.

### **2.5.1 The number of IDM providers will increase**

HP surprisingly withdrew from the identity management market in February 2008. This shows that competition in the identity management market is so big that it even poses problems for the giants on the market. There are mainly small suppliers in the identity management market whose core field of competence is identity management or just a part of it. Apart from HPs' withdrawal, the trend goes towards the opposite direction. More and more suppliers will enter the identity management market as identity management is increasing and therefore becomes financially lucrative from the suppliers' point of view.

The Radicati Group has carried out a study according to which the market in the identity management field will keep growing strongly. Thus in 2007 the market comprised worldwide incomes of 2.8 billion US dollars and will grow to nearly 13 billion US dollars until 2011. The Forrester Group however foresees 12.3 billion US dollars until 2014. Looking at these figures it is not surprising that more and more suppliers are entering the identity management market.

Furthermore, especially the development of identity management towards openness and modularity will give new suppliers the chance to enter the market. (Radicati 2007; Cser 2008; Penn 2008)

### **2.5.2 Companies will use federated identity management**

As already mentioned, federated identity management takes place at a general level, i.e. between companies. Such a system requires a special trust relationship, as users have to pass the entire information on to a single institution. This is a further step towards mass surveillance. Passport was a trial by Microsoft to install a central authentication service; it failed due to a lack of trust. Too few users were prepared to entrust the Microsoft Corporation with the entire data about themselves. Therefore, only parties that are generally trustworthy are able to play the role of identity providers, i.e. the state or banks.

These examples show that virtually nobody would transfer their entire data to a single party. However, users are generally prepared to entrust several different institutions with parts of their data. The typical federated identity management approach therefore is decentralised where users can choose between several identity providers (IdPs). (Hommel 2007)

The prerequisites for a federal identity management are standards, technologies and a basis of trust, also called Circle of Trust. Technologies executing this standard are for example Security Assertion Markup Language (SAML) or the Liberty Alliance Framework. The SAML standard marks an expansion of identity federations around a central coordination service, which decreases the initial implementation effort and improves scalability. (Mezler-Andelberg 2008)

### **2.5.3 Privacy and data protection will be gaining importance**

In the field of identity management, the topics of privacy and data protection will gain higher importance. Thus, authentication systems like OpenID have been in discussion in the past because of the ease of phishing attacks on the systems. The problem is the following: when a service is requested, the service provider can have malign intentions and send the user to a faulty identity provider. The user will leave their credentials there, thus exposing them to attackers.

There are several more problems regarding the privacy of Single Sign On users. As users log on to websites through an identity provider, the identity provider will store a list of the visited websites. Therefore, the identity provider can see which websites someone uses every day. This list can also be used by potential hackers who could access protected and digitally transferred user data, like e.g. passwords or credit card numbers. (Kuppinger 2008)

#### 2.5.4 Identity 2.0 will be the base of future IDM systems

According to the analyst Group “Kuppinger Cole and Partner”, the top trend in the field of identity management in the year 2008 is the so-called Identity 2.0, named so for marketing reasons derived from Web 2.0. Because of the increasing development of IT and networking in the past years, the amount of digital identities also gained enormously. This increase is not limited to the commercial area, but goes beyond it and extends to the private sector.

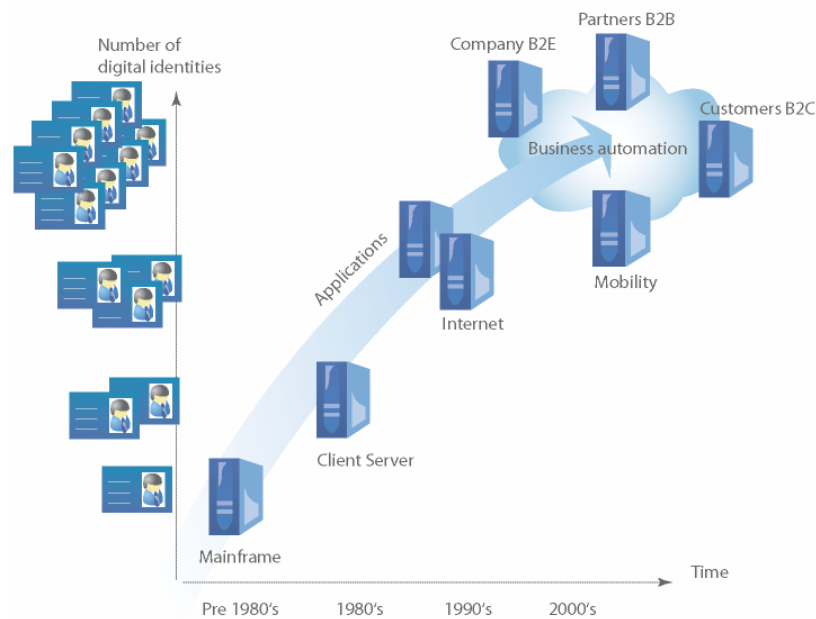


Fig. 2-2 Increase of digital identities. On the basis of (Lukawiecki 2006)

The term Identity 2.0 is a synonym for user centric identity management (UCIM). In view of this term, it already becomes evident how Identity 2.0 differs from IAM or in-house IDM. It is the user that is in focus. This also constitutes the main difference to Identity 1.0. Within Identity 1.0 not the user, but the website is in focus; this is why Identity 1.0 is sometimes called site-centric.

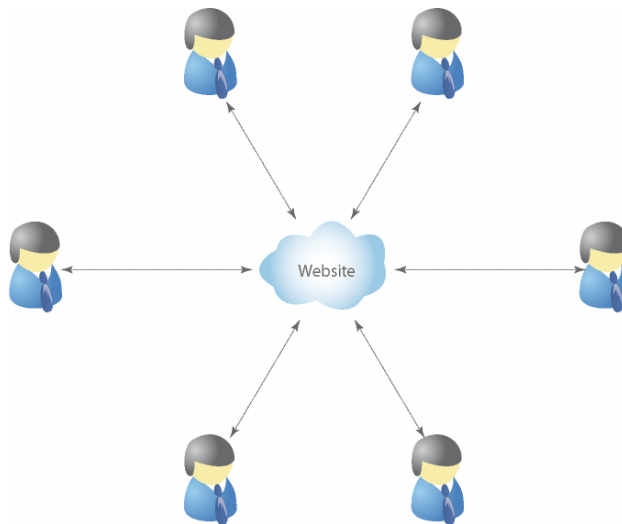


Fig. 2-3 Identity 1.0 is site centric. On the basis of (Hardt 2005)

Here, users do not have the possibility to decide who will be able to access their data and who will be denied access. Within Identity 1.0, identity is not described by the information the user provides on the website, but by the information which the site itself stores in the course of time about the person. As this information is only known to the site, it cannot be transferred to other sites by the user. The websites therefore act as so-called “silos”, i.e. they store information just for themselves, but they are not able to communicate this information with other sites.

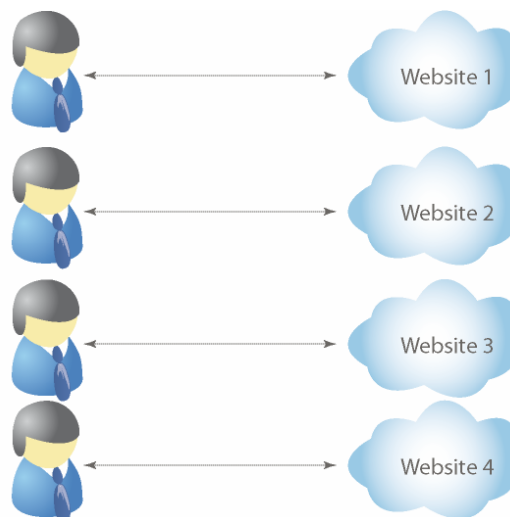


Fig. 2-4 Identity 1.0, on the basis of (Hardt 2005)

As the above figure shows, the user is a member of several sites, but he cannot transfer his data from one site to the other, as the data is stored and administrated only locally by the respective sites. In Identity 2.0 however, the user stands in focus. The principle of informational self-

determination is applied here, so every user can decide which data is published and distributed; the principle of privacy concerning data is respected. There are three main functionalities of a user-centric identity management system: the administration of partial identities, the protection of privacy and the safety of identity data. (Hardt 2006)

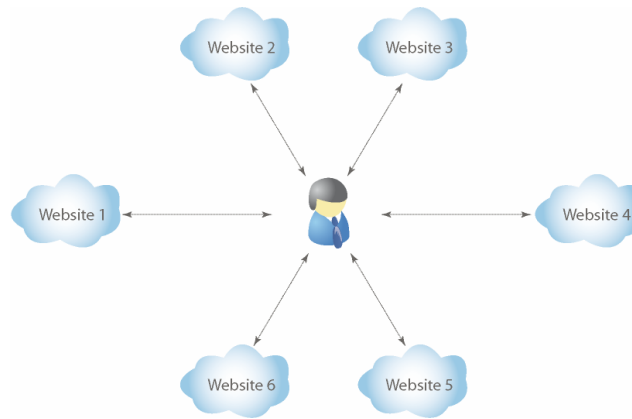


Fig. 2-5 Identity 2.0, on the basis of (Hardt 2005)

### 2.5.5 Biometrics will contribute to increase the security of IDM systems

Experts anticipate that biometrics will have a growth spurt in the identity management market. The demand for such solutions is foreseen to grow about 15% in average within the next three years respectively. (IT Reseller 2008) With regard to identity management, the application of biometrics at the moment mainly concerns authentication. The application of biometrics solves the identity problem and determines the person trying to authenticate himself. The identity management system then takes control over the respective identity's rights. Biometrics is mostly applied during authentication, when more security is needed than the application of passwords can supply. Seeing the progress in the biometrics sector it is not surprising that this kind of authentication also plays a role in identity management. Additionally, it increases the company's safety with regard to former employees as their inoperative user accounts still contain their biometric characteristics. In the case of an attack of a former employee on the company the biometric characteristic will show who the attacker is.

One can also observe the development towards biometrics on the IDM market. Siemens is the first supplier to offer a complete solution consisting of identity management and biometrics. (Siemens 2008) Since 1<sup>st</sup> June 2008 "Identity Management and Biometrics" has been available on the market on which Siemens will try to grow above-average in the future. Siemens anticipates a growth of the package of around 20% per year. (Roggenbuck 2008)

## 2.6 Evaluation

The identity management market is developing rapidly. The reasons for the increasing investments of numerous companies in this industry are obvious. In the future, the investments in this market will rise further; as a consequence the amount of providers will increase even more. The direction of development in this sector is driven by security and performance aspects. In the future, big companies will mainly invest in modular parts of IDM technologies compatible with their present systems. Furthermore, security will increase through authentication and authorization.

The biggest trend in identity management can be foreseen in the global IDM, which can be separated into federated and user centric IDM. The user centric type of IDM, however, is not very common, mainly because of the phishing problem. If a user loses his password, while applying a user centric IDM system, an attacker will be able to access all data of the legitimate user.

The expected growth of biometrics in identity management systems, otherwise only established in high security applications, shows the importance of biometric identity.

## *Chapter 3*

### **3 BIOMETRICS**

---

The previous chapter has shown the importance of biometrics as a new trend in identity management. However, it is important to understand the principles on which biometric systems work in order to use them. As an example for biometric systems, this work concentrates on typing behaviour. Psylock, the biometric method for typing cadence developed at the University of Regensburg, provides very good user recognition and is one of the few biometric methods that can work in the web.

---

#### **3.1 Motivation**

The constant development of the World Wide Web, particularly since Web 2.0, makes increasingly higher demands on data protection. A new way of protecting access to personal areas is the use of biometric authentication methods. There are several features in a person that can be used for clear identification. Today, we frequently encounter security systems based on fingerprints (Jerra 2008). Retina scan as well has been implemented in many security systems and is used in practice (Merl 2007). One of the main problems is the acquisition of biometric features, as it requires additional hardware and installation. This does not only cause high costs and effort of implementation, it is also hardly pleasing the users. User acceptance, however, is a decisive factor in the purchase and the integration of biometric systems.

The University of Regensburg developed a biometric authentication system based on the typing cadence of a person, i.e. the way he types at a computer keyboard. Years of research and field tests proved that every person is unambiguously identifiable by his typing features (Psylock 2008). The system extracts and classifies these features and creates a reference pattern representing the typing cadence. A computer keyboard serves as sensor for recording all necessary data. User acceptance is good and the costs of the system are low, as there is no need for additional hardware.

As in all biometric systems, however, typing cadence exposes the effect of a worse user recognition rate in the long term. The reason for this lies in the fact that biometric features are subject to a process of change. The natural aging of a person can change her epidermal ridges or her retina structure. While the aging takes many years for so-called morphological features, dynamic features such as the typing cadence age much faster. Not only the aging of a reference pattern, but also fluctuations in the daily shape, injuries etc. can strongly influence the typing cadence and make great demands to a system to adjust to these changes.

This chapter gives an overview of the concepts, terms and methods used by biometrics on the example of typing cadence.

### **3.2 Terminology**

Biometrics:

Biometrics are methods for measuring persons, used in communication and information technology. In information security, biometric methods try to identify an individual by his biometric features. Biometrics includes all physiological features such as retina, fingerprints, vein structure or hemogram, as well as behavioural or movement features such as gait, signature dynamics and typing cadence. It is very well apt for identification, because it can not be lost or stolen, as opposed to passwords or keys. A person's features are unique and can be assigned to one person only. Biometric technologies are defined as „automated methods of verifying or recognizing the identity of a living person based on a physiological or behavioural characteristic“. (Monrose 1999; Rubin 1999)

These technologies are increasingly popular, as they can add to the security of a system when combined with other well-proved techniques.

In this work, biometrics is only regarded as the automated identification of persons based on their physical features. (IT Lexikon 2008)

Biometric sample:

A biometric sample is an analogue or digital representation of a biometric feature before the process of feature extraction, which is created by a biometric data capturing device or a biometric capturing subsystem. An example would be an electronic passport photograph.

A biometric sample is delivered by a sensor, which is the main component of a data capturing device. The sample usually consists of more information than necessary for a biometric identification, it is raw data. In many cases, as with the photograph, it is the direct image of a biometric feature. (Bromba 2008)

Biometric template:

A biometric template is a particular biometric reference, which saves biometric features for the purpose of comparison.

The comparison takes place during the identification process, comparing the saved biometric template and the current biometric characteristics, which were gained from raw biometric data delivered by the data capturing device or sensor. This process includes the calculation of a match rate that states in which measure the sample corresponds to the template. (Bromba 2008)

Enrolment:

The process of enrolment creates a reference pattern (template), which serves as a starting point for authentication. During this process, a person delivers several biometric references, from which a characteristic template is created by means of feature extraction.

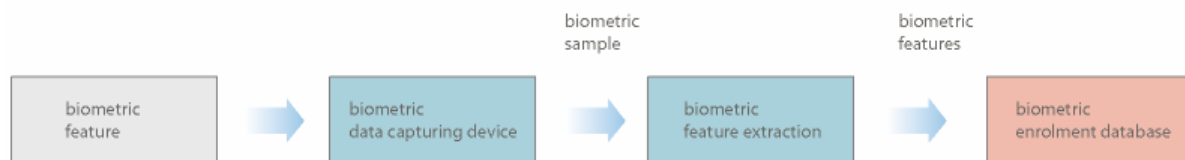


Fig. 3-1 Typical internal enrolment process (Bromba 2008)

On the example of typing biometrics, the enrolment process is conducted in the following way: the keyboard sensor records the key events and stores them locally in the computer in the form of a typing sample. This sample is then submitted to a server, where a biometric component collects several exemplars. After a predefined number of samples, the enrolment process ends with the creation of a biometric template and the calculation of a user profile. A user can have several profiles, e.g. for different sensors.

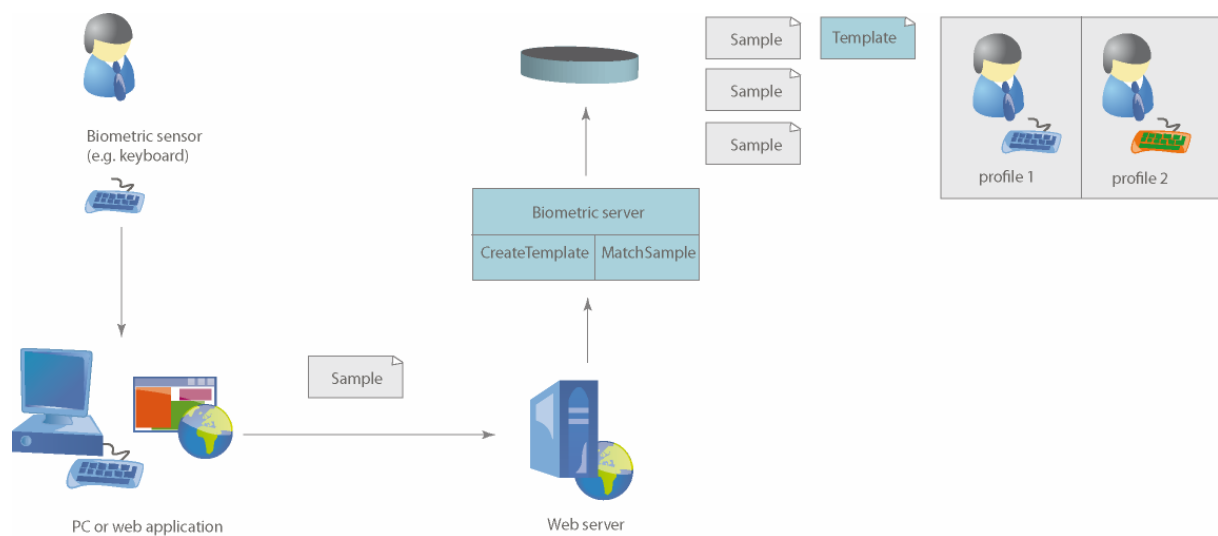


Fig. 3-2 Functionality of biometrics

#### False acceptance rate (FAR):

A false acceptance rate is the probability with which unauthorised persons are taken for authorised during authentication. This rate is very relevant for the security of a system. The higher the FAR, the more possible is a successful attack on the system. Biometric systems require a minimum threshold that the match rate has to achieve in order to be authenticated. The higher the threshold, the more secure is the system and the lower is the false acceptance rate. (Bromba 2008)

#### False rejection rate (FRR):

The false rejection rate is the probability with which the authorised user is taken for unauthorised and denied access. The higher the safety requirements to a system are, the higher is the threshold that needs to be achieved by the match rate. This, however, increases the number of false rejections, which has a decisive influence on the user acceptance, as an authorised person has to authenticate several times one after another to be granted access to his system. (Bromba 2008)

#### Equal error rate (EER):

In the ideal case, a biometric system would have the FAR and FRR of 0%. In practice, this is not possible, which makes it necessary to examine the correlation of the two error rates. As described above, a pre-defined threshold decides about the security level and the user acceptance of a system. Depending on this value, the number of falsely accepted and the falsely rejected users changes. The point of intersection of both rates is called the equal error rate. This value is used to define the lower limit of a system's optimisation. (Bromba 2008; ITWissen 2008; Uni-Magdeburg 2005)

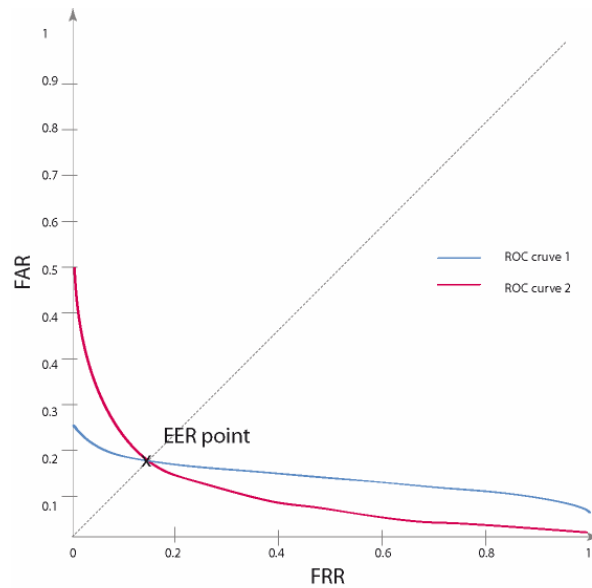


Fig. 3-3 FAR/FRR curve

An efficient biometric identification system has to be able to function with a high security level, or threshold, without rejecting too many authorised users.

Failure to Enrol (FTE):

The failure to enrol rate shows the number of persons that were unable to go through the enrolment process, be it for physiological reasons or because their features were not clear enough to be identifiable. If this value is too high, the biometric method is not apt for the use with large numbers of people. (Bromba 2008)

Biometric identification:

The principle of biometric identification is the following:

Upon enrolment, the user delivers several biometric samples and thus creates a template. When the same user wants to authenticate, she delivers another sample at the data capturing device. From this sample, the features necessary for comparison are extracted and classified. This information makes it possible to make a comparison between the sample and the template. The result of this comparison is the match rate, which shows the measure of similarity between the two values in percent. If the match rate is higher than the threshold pre-defined by the system, the user will be successfully authenticated.

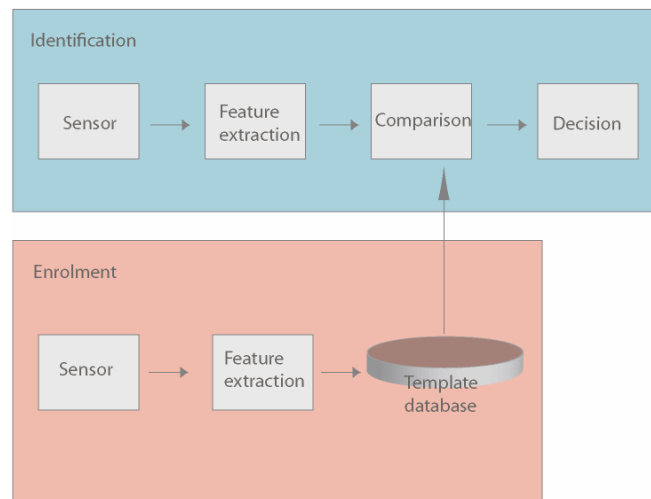


Fig. 3-4 Identification and enrolment process (Pike 2008; Bromba 2008)

Attacker data:

In order to create an unambiguous profile of a user's typing cadence, the system uses not only the person's own data, but also so-called attacker data. This attacker data represents the typing cadence of other users. Using this together with the user's own data, a profile can clearly distinguish between the user and the rest. Based on attacker data, the features can be weighted user specifically. If a feature is particularly different from the rest population, it will be weighted stronger. The overall identification rate of a biometric system is strongly ameliorated by this measure. (Omote 1999)

Identity:

The term of identity describes the unique combination of personal and unmistakable characteristics of an individual. By these characteristics, the user of a system can be clearly and definitely distinguished from others. An identity check is possible with biometric methods. (Blöckenwegner 2006)

### 3.3 Typing cadence as a biometric method

Worldwide, a handwritten and signed document is recognised as a legally valid declaration of intent. Every person has her own style of signing documents. Signatures differ in the pressure and the drive of the writing hand. In an analogue way, the way a person types at a common computer keyboard can give valuable information about the typist. Every person has his own speed, rhythm, constancy and precision of typing, which can be measured and analysed. These features could be used for identifying a person. For this, it would be necessary to have a method which is able to

extract and classify biometric characteristics from a typing sequence. This would be necessary in order to create a reference pattern that would enable biometric identification, as with all other biometric methods. (Breu 2001)

### 3.3.1 Classification of typing cadence biometrics

Generally, a person's biometric features are driven by three factors. All three contribute to the final resulting feature in different measures. These three factors are:

- Genotypic: features that are mostly defined by the structure of a person's genetic information.
- Randotypic: features shaped in the early stage of embryonal development.
- Conditioned: features that a person learned through constant repetition and execution and that are developed in time.

The following table estimates the weighting of different biometric characteristics:

Biometric feature	genotypic	randotypic	conditioned
Fingerprint (only minutia)	○	○○○	○
Signature (dynamic)	○○	○	○○○
Face geometry	○○○	○	○
Iris structure	○	○○○	○
Retina (Blood vessel structure)	○	○○○	○
Hand geometry	○○○	○	○
Finger geometry	○○○	○	○
Vein structure of the hand	○	○○○	○
Shape of ear	○○○	○	○
Voice (Sound)	○○○	○	○○
DNA	○○○	○	○
Smell	○○○	○	○
Typing cadence	○	○	○○○
Comparison: Password			(○○○)

Table 3-1 Classification of biometrics after (Bromba 2008)

(○ = low, ○○ = medium, ○○○ = high)

As shown in the table, typing cadence is a mainly conditioned and at that an active and dynamic feature. (Bromba 2008)

### 3.3.2 Criteria for biometric features

In order to compare typing cadence with other biometric methods, it is necessary to first define some basic criteria to decide over the efficiency of a system. At first, the biometric feature at hand has to be scrutinised for its aptness. A biometric feature has to meet the following conditions:

Universality:

A biometric feature should appear at every person. Nevertheless, there is the possibility that some persons will not or not sufficiently show certain features. A study confirms that Asian women have trouble delivering an adequate fingerprint due to their small hands and flat epidermal ridges. Therefore, they can not enrol to the system successfully (Failure to enrol rate). (Pugliese 2007)

Uniqueness:

A feature has to differ sufficiently between persons as to ensure the clear identification of an individual. This is particularly strong with randotypic features that have been set up at an early embryonal stage, such as iris or fingerprint.

Permanence:

Permanence describes the measure in which a feature is subject to changes over a longer period of time. Small changes can easily be absorbed by making adjustments to the pre-defined tolerance area. A further dilution of this problem is achieved by adaption, which constantly updates the basic template. Permanence is an important issue for typing cadence because of its dynamic character.

Biometrics	Universa- lity	Unique- ness	Perma- nence	Collecta- bility	Perfor- mance	Accepta- bility
Face	ooo	o	oo	ooo	o	ooo
Fingerprint	oo	ooo	ooo	oo	oo	oo
Hand geometry	oo	oo	oo	ooo	oo	oo
Keystrokes	oo	oo	o	ooo	oo	ooo
Retinal scan	ooo	ooo	oo	o	ooo	o
Signature	o	o	o	ooo	o	ooo
Voice	oo	o	o	oo	o	ooo
DNA	ooo	ooo	ooo	o	ooo	o
Gait	oo	o	o	ooo	o	ooo

Table 3-2 Comparison of various biometric technologies, modified from (Jain 2004)

(o = low, oo = medium, ooo = high)

### 3.3.3 Criteria for biometric methods

Not only the choice of features is dependent on certain parameters, the technical implementation as well has to follow certain quality criteria. A biometric authentication system has to have the following characteristics:

#### Acceptability:

Acceptance of the biometric method is an important criterion. If the users are uncomfortable with showing their feature, a method based on this feature is definitely not usable in practice. The fact that biometrics would render PINs and passwords obsolete is generally approved in public. Nevertheless, many people are sceptical. Especially in private life, hardly anybody can imagine using biometrics instead of the present technologies. The decisive factor is often the question of the ease of enrolment and authentication. If this can be done in an uncomplicated way at the workstation, the probability of a good acceptance is higher. If the users are uncomfortable with the authentication, they may refuse using it or try to circumvent it and thus render the authentication useless. A further important aspect is data security. The use and storage of biometric data should be transparent for the user, so that fear of misuse does not occur. Additionally, the system should work stably and reliably in order to avoid frustrated reactions on the user's side.

#### Performance:

Three factors are very important in the way of performance of the system. The first is its accuracy, which is a requirement to the method itself. The authentication should take place quickly in order to process large numbers of people, e.g. at the entrance to a company building. Last but not least, it is exigent that the system should be very robust so as to avoid times of breakdown and therefore forced non-authentication, which could prove damaging to the productivity of employees. "The biometric method has to be implemented in a technical system that works quickly and uses few resources." (Bakdi 2007)

#### Circumvention:

Especially secured systems are often the aim of malevolent attacks, as they usually protect valuable data. The biometric technology used to protect such data has to be stable and secured against all possible kinds of fraud attempts. (Amberg 2003)

Collectability:

The cost of development and use of a system based on a biometric method have to be in a reasonable correlation with its benefits. These benefits prevail only in the case that the system is well accepted by its users and when the resistance to circumvention is justifiably high. (Amberg 2003)

### **3.3.4 Particularities of typing cadence**

The advantage of typing cadence lies mainly in its easy collectability. While all other biometric features need to be captured with a special sensor, typing cadence only requires a computer keyboard. There are several methods to measure typing cadence. Some of them rely on the pressure that is made on the keys; some only take into consideration the times of pressure and release. The pressure-based methods, however, require a particular type of keyboard and therefore a sensor like all other biometrics. The methods that go without special sensors have the advantages of higher mobility, easier maintenance and lower acquisition cost. A user is able to authenticate from any computer with a keyboard that has an interface with the program API, e.g. a web access.

Another particularity is the strong fluctuation of the feature. Typing cadence is, unlike fingerprint, mainly under the influence of the user. This is why the samples differ much more than with morphological features which may be subject to measurement imprecisions. The challenge is to conceptualise a stable system that can authenticate users notwithstanding the fluctuations in the day's form. (Bartmann 2007)

A further benefit of typing cadence is the fact that it is a behaviour that the users encounter in their daily life and that is accepted and applied by them rather than other methods that require them to act out of the ordinary. The user authenticates to the system by means of his keyboard, the only difference to the password being that the system gives him a predefined sentence to type.

### **3.3.5 Operational areas**

Typing cadence has the usual benefits of biometrics towards knowledge-based methods of authentication. It can not be lost, stolen or passed on to other persons. As opposed to morphologic features whose operational areas lie mostly within criminology and person identification on border crossings, typing cadence is most apt to secure the access to logical IT systems. Typical use cases are the user authentication at the workstation or the protection of web applications. (Peacock 2005; Tapiador 1999)

If an enterprise outsources the maintenance of its IT to another company, it often has no control over which employee of the IT service provider is responsible for an action or a mistake, as there is often only one account or the data to their personal account is passed on to colleagues. Typing cadence can determine who exactly typed a sequence and therefore who performed a certain action.

Paid online services often face the problem that the access data for their services is passed on to family members or friends of their users. With typing cadence analysis, it is possible to restrict this practice so that users can not share the same account anymore.

Another possible use is the combination of typing cadence with other authentication methods, creating so-called hybrids. (Anagun 2002) proved in his research that the combination of typing biometrics with voice recognition offers excellent results. In combination with knowledge-based technologies, it is possible to enhance passwords instead of replacing them, thus increasing their security. Upon typing, the typing cadence is checked as well as the correct password. It is also possible to reduce the time and personnel expense by binding password reset to biometric authentication. (Bakdi 2007)

### **3.3.6 Typing cadence by Psylock**

Psylock, a biometric system for user recognition by typing behaviour, was developed by the University of Regensburg and is based on statistic methods and methods of artificial intelligence. The event of typing is assumed to be a random experiment and is modelled with adequate distribution functions. The estimators and the derived and classified characteristics generate a reference template. This template is the basis for further comparisons with the later samples.

As most typing cadence biometrics, Psylock analyses the length of pressure, the rhythm and the speed of typing. However, these characteristics fluctuate depending on the day's form of the user and are not sufficient for a stable authentication. The particularity of Psylock is that it uses additional, stable characteristics inherent to typing cadence. These characteristics are psychometric features calculated by mathematical inference methods. An example for these characteristics is the use of Shift keys, or the so-called overlaps, the cases when a second key is pressed while the first was not yet released. In order to increase the quality of the identification, Psylock uses attacker data that help enhance the differences between the typing cadence of an authorised user and another person. The system delivers excellent values and is comparable to other biometric methods in its stability and security.

The quality comparison is shown in the following graph. The graph shows the false acceptance rate and the false rejection rate of Psylock with different text lengths in comparison with other biometric methods. (Breu 2001)

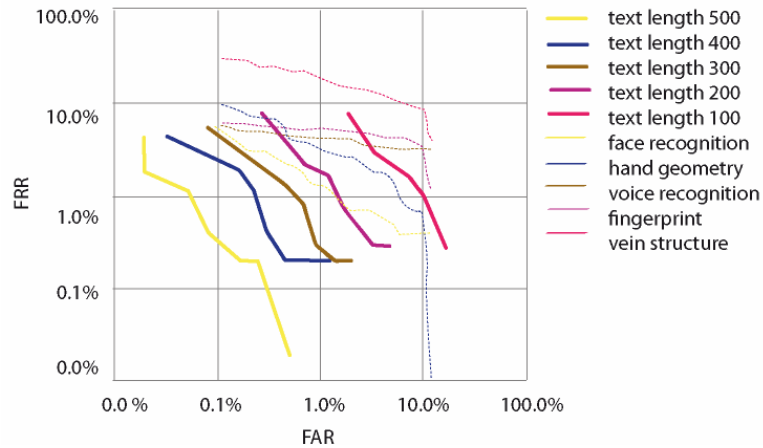


Fig. 3-5 Psylock in comparison to other biometrics (Centre for Mathematics, 2002)

Tests have shown that the more text a user types, the more data about his typing cadence he delivers to the system and the better is the quality of identification. A long input text, however, would lessen a user's acceptance of the method. In order to shorten the enrolment and make it more comfortable and in the end more acceptable for the user, Psylock uses a predefined sentence instead of an arbitrary text, which allows to clearly identify a user after very few characters typed.

The use of a predefined test sequence has two big benefits regarding user acceptance. The first is the reduced fear of surveillance by the employer, as the data collection is transparent and takes place at a certain moment, instead of operating in the background without the user's knowledge. Secondly, the user acceptance is higher if the input sentence is kept short. The disadvantage of the system is that upon enrolment, the number of repetitions necessary for the training of the system is increasing with shortening the sentence. The shorter the input text, the more typing samples are necessary in order to create a significant reference pattern. (Bakdi 2007)

### **4 AUTHENTICATION AND AUTHORISATION INFRASTRUCTURES**

---

AAI systems regard the Single Sign On component as “the holy grail of authentication” (Smith 2002). By Single Sign On, the local passwords from various sites are replaced by a single authentication, far more comfortable for the user. In order to improve the authentication process, an AAI has to be selected that will be the base for a prototype developed during this work. The selection and the possible AAIs are presented in this chapter.

---

#### **4.1 Definition and role of AAI**

The login procedure is an important part of portal solutions and interactive web applications. Through the login process usually based on a password chosen by the user, the user identity is to be determined and the access to some protected or personalized content to be granted.

The disadvantage of many current web and internet applications consists in the fact that they offer individual solutions for the login procedure and user management, thus making it necessary for the user to register to each application and then to manually login to each one of them. This redundancy in the input of user data is not only less user friendly, it also presents an important security risk, namely the fact that the users are forced to remember a large number of username and password combinations. A study made by the company RSA (RSA 2008) shows that 36% of the experienced internet users are using 6 to 15 passwords and 18% of them even more than 15. From these numbers, it is obvious that it is difficult to manage such a big number of user data. In this case, users have the tendency of using simple passwords, of writing them down or simply using the same password everywhere.

Another disadvantage on the side of the service provider is the fact that he has to develop and maintain his own login procedure. This is reflected in additional work and costs but can also present a security risk when the login procedure is not properly implemented.

The data which users store on different servers is also subject to aging, for example when the users change their residence, they will not manually update the new address everywhere.

The purpose of Authentication and Authorisation Infrastructures (AAIs) is to provide an authentication system that is designed to resolve such problems. The AAIs are a standardized method to authenticate users and to allow them to access distributed web contents of different web providers.

## **4.2 Requirements analysis**

In order to implement a biometric AAI, several AAIs must be studied in order to compare their similarities, concepts, advantages and disadvantages. The following set of requirements has to be met by the AAI chosen for biometric integration:

- User-centric identity: The AAI must follow the principle of user-centric identity. This principle considers the user as the centre of the system and allows him to control and determine which information should be given to which service and even makes it possible for him to revoke this data.
- Open source: For the development of an additional biometric component for the AAI, it has to be open source. Furthermore, as the application will manage critical user data, it is also necessary that the AAI is not the proprietary platform of a company, thus avoiding possible “security by obscurity” problems.
- Clearly defined standards: Another important criterion is the interoperability of the AAI with other systems of the same type, therefore it is necessary to have clearly defined standards and interfaces in order to avoid producing a custom solution which can be used only in small closed circles.
- Possibility of customization: The necessity of having a solution which can be customized in different ways may sound contrary to the aforementioned criteria. In reality there is no contradiction between clearly defined standards and the possibility of customization: the AAI standards should have clearly defined components and interfaces, but within these to allow actions like the changing of encryption algorithms or of the authentication technology.
- Possibility to expand to biometric systems: In connection to the previous point and to the future applications, it is desirable to have a solution that is meant to be extended with biometric login without leaving the AAI standard.

- Implementation effort: The chosen AAI has to be interesting not only for big web providers, but also for small web services. The implementation effort for the chosen solution should not be big, as small companies could not afford to use it.

- Existing frameworks: In order to minimize the implementation effort and also to avoid creating non-standard solutions, the AAI should already have reference implementations or frameworks. These should be available in several programming languages.

- Future compliant solution: In the web, there are many AAI systems. Some of them were made as special solutions for limited operation fields, some others were meant as universal concepts for portals, intranets or small web services. Some of them had a very promising start and then failed or their development process was no longer continued; other AAIs were never very popular and at others it is impossible to predict the future development at the moment.

Therefore it is an important criterion that the chosen AAI should be maintained by a large software developer community in the close future. However, this criterion is difficult to quantify. Clues to it can be the number of implementations already made, the actuality of used technologies as well as the size and activity of the developer community and the companies that promote the technology.

### **4.3 Basic concepts of AAI systems**

#### **4.3.1 AAI components**

The tasks of an AAI system are primarily the authentication and the authorisation of a user. At first, the identity of the user is checked during the login phase: here it is proved whether the user of the systems is really the one he claims to be, which happens most frequently by means of a password. If this question is answered positively, the next step is the authorisation, which allows the user to access protected web services that are defined by the user rights management of the system.

The construction of an AAI can be characterized by four components (Schläger 2007):

- Single Sign-On (SSO): This refers to the ability of the user to access the services for which he is entitled after a single authentication on all computers, without the need to register again.

- Attribute Assertion: This allows the exchange of user attributes between different parts of the system, so that the user attributes are redundant.

- Policy Decision: This decision is based on the user credentials and user attributes; it states whether the user should be granted access to the requested resource.

- Policy Enforcement: Enforcement of the policy decision, e.g. error message or redirection to the resource.

#### **4.3.2 Ticket systems**

A similarity of all researched AAI is the use of so-called online tickets for the authorization process. In this process, after a successful login the user receives an encrypted and signed ticket, which documents that he has previously logged in to a component within the AAI successfully.

If the user wants to access other resources, he does not have to log in again, but merely shows the previously received ticket to the online resource. This process usually happens in the background, so that after a single login no further actions are required from the user in order to access other protected areas.

From the user's perspective, it does not matter whether the registration process exchanges online tickets or credentials between the system components, since this operation is invisible for him as it runs in the background.

The advantage of online tickets consists in the fact that sensitive data such as passwords is being transferred only once. However, it is possible for an attacker to capture such a ticket by means of a "man in the middle" attack and to use the ticket afterwards for his own purpose.

To avoid this or to make it more difficult for the attacker, online tickets are usually furnished with a timeout, therefore the time frame for a possible attack is reduced. The tickets contain unique random numbers in order to avoid a replay attack, e.g. the use of the same ticket several times.

Two different models can be used within the ticket system, the circle of trust and the central system approach.

#### **4.3.3 Circle of Trust**

The model of the Circle of Trust is a way to realize a Single Sign On based on online tickets. This is characterized by the union of equal resources. *Equal* in this context means that the user can log in to each application with his password or with biometrics. Then he receives an online ticket, which he can use to access several resources without the need to register again.

The exchange of online tickets can be made by using cookies or URL rewriting. The first possibility has the problem that browser cookies can be disabled. The second assumes the existence of a return address that would grant the user's request for further resources by means of an online ticket.

One difficulty in using the ticket approach model is the fact that the corresponding applications have to generate compatible tickets in order to ensure the interoperability of components.

The main advantage of the Circle of Trust lies in the security of its architecture, because the user does not handle previously issued tickets; these are generated only upon log in to a certain resource. This advantage, however, is connected to an increased implementation effort as each component must implement its separate login procedure. In addition to that, the user needs to register separately to all components.

#### **4.3.4 Central Single Sign On server**

The principle of the central Single Sign On server provides a cleaner system approach, where a dedicated server is responsible for the login and the generation of online tickets. Seen from the point of view of the logic flow, the user can begin by logging in to the central server and then choosing between different resources he would like to access.

However, it is also conceivable that the user first accesses an application and then, if he has no ticket, the application asks the central server for permission. After a successful login, the user is bounced back to the original resource, which will not ask him to log in but merely check the ticket he received before from another authority. (Rummeyer 2006):

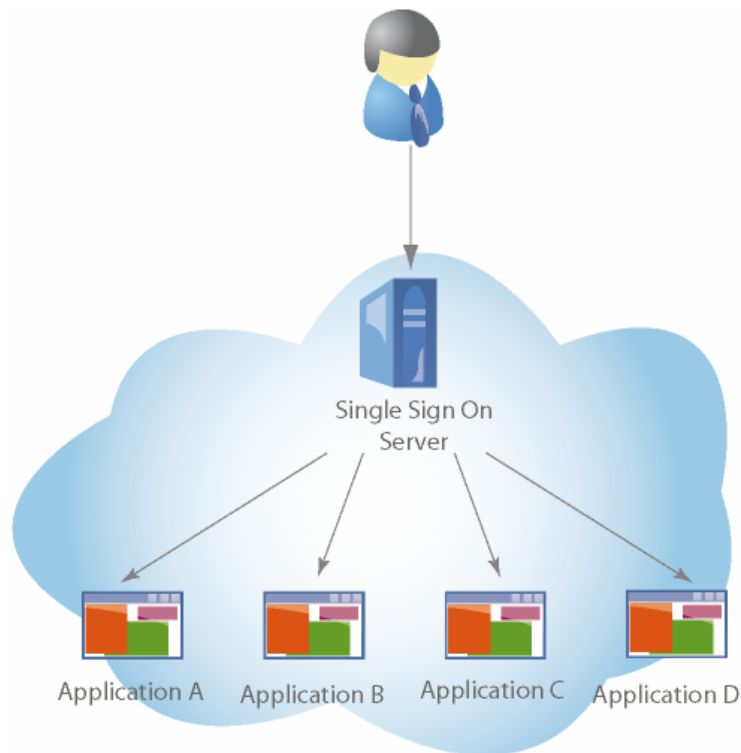


Fig. 4-1 Single Sign On

The advantage of this model in comparison to the Circle of Trust lies in the fact that a lower redundancy is achieved through a smarter division of tasks such as the login process and the ticket generation process. Its disadvantage consists in the fact that the central Single Sign On server has lower system reliability: if the dedicated server is down, it is impossible to login to any of the components (single point of failure).

#### 4.4 Considered AAI systems

Before the implementation of a biometric AAI, several solutions available were considered and their capabilities compared with the criteria of the requirement analysis.

##### 4.4.1 Central Authentication Service (CAS)

Originally developed by Yale University, the CAS project represents a relatively simple approach to realize a Single Sign On. It follows the centralized approach of the ticket systems, which makes a clear division between the client and server. For both client and server components, official reference implementations exist in Java as well as other programming languages. Technically, CAS is based on various open-source technologies, including the Spring Framework.

The system impresses mainly by its relatively low complexity and the possibility to easily integrate in existing applications. An implementation of biometrics would be also conceivable due to the modular design of the system.

However, CAS cannot be used for the purpose presented here as it is limited exclusively to the Single Sign On and provides no opportunities to manage user attributes.

#### **4.4.2 Shibboleth**

Shibboleth is a central ticket system developed by the Internet2 consortium and is particularly wide spread at universities and other educational institutions. The reason may be that Shibboleth allows a very detailed description of user rights, which are necessary in complex organizations.

Technically Shibboleth is based on the SAML standard, an XML language for defining access protocols. Freely available implementations of Shibboleth exist in several programming languages. Its architecture consists of three parts:

- Identity Provider: This part is responsible for the generation of tickets and user registration. Each user has only one identity provider which is located in his hometown organization, such as his university. However, multiple identity providers which can issue tickets may exist.
- Service Provider: The resource that the user wants to access.
- WAYF (Where are you from?) server: An optional server in which the user selects his own organization. Also called localization service.

The process of Shibboleth authentication consists in the following steps: (Swiss Education 2007):

1. The user wants to access a certain resource, such as online courses at his university. The resource realises that the user does not have a ticket.
2. In this case, the resource redirects the user to the central WAYF server.
3. The WAYF server shows an input mask to the user, where he can choose his home organization.
4. The user chooses his university and sends the form.

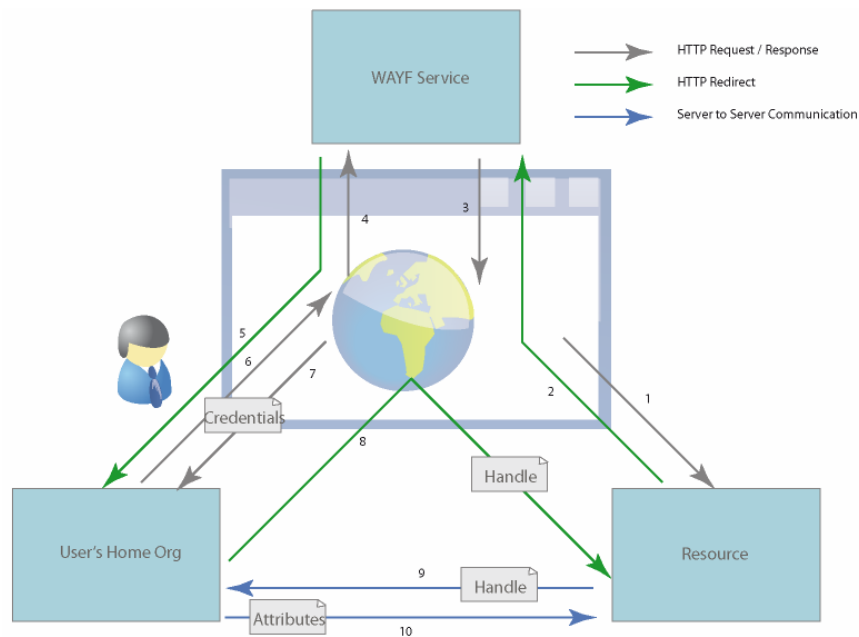


Fig. 4-2 Shibboleth architecture (Swiss Education 2007)

5. The WAYF server redirects the user to his home organization.
6. The home organization shows a login form to the user.
7. The user inputs his credentials and submits the form.
8. After he is successfully authenticated, the user receives a ticket and is redirected to the originally requested resource, to whom he shows a ticket (e.g. per URL rewriting).
9. The resource contacts the home organization in order to verify the ticket.
10. If the home organization recognizes the ticket, it will send the user attributes to the resource, which will compare these with its user policies and allow the user to access the online content.

The advantage of Shibboleth is, as already mentioned, the ability to define detailed access policies. Due to the use of the SAML standard, Shibboleth is future compliant in terms of compatibility and security. However, a major disadvantage is the high complexity of the system, so that Shibboleth seems hardly eligible for smaller Web applications, despite existing reference implementations. The main exclusion criterion for Shibboleth lies in the fact that it contradicts with the principles of user-centric identity: the network assumes equal trust relationships between the organizations at institution level, so that the user has no possibility to control his data, nor to configure his own trust network.

#### **4.4.3 Liberty Alliance**

The Liberty Alliance project is a branch spreading industry standard, which was initiated in 2001 by Sun Microsystems as an alternative to the Microsoft Passport system. Liberty Alliance is based like Shibboleth on the SAML standard, but pursues the concept of a decentralized ticket system. The Liberty initiative covers 150 considerable IT manufacturers (HP, Intel, IBM, Sun, Novell) at the moment, which is important for future security.

The experience of a preceding project (Pernul 2006) with Liberty Alliance, however, proved that the standard is very inaccurate in crucial points which make a compatible implementation almost impossible.

Despite the lobby behind the project, there is only little activity from the developer community, which can be seen in the few reference implementations available. These experiences are affirmed by the fact that, at the moment, there is only one freely available Liberty Alliance implementation of SourceID. This implementation contains only the basic functionality which Liberty Alliance presents in theory. Due to these negative experiences and the assumption that Liberty Alliance standard is not fully future compliant, a biometric implementation in Liberty Alliance is not possible at the moment.

#### **4.4.4 Windows CardSpace**

After the centralized Passport system did not get general acceptance, the CardSpace project represents a further attempt of Microsoft to establish an AAI.

CardSpace is an identity management system developed by Microsoft which is integrated in the operation system since Windows Vista. CardSpace permits, like OpenID, a decentralized administration of digital identities.

The card component is taken out of real life and it is meant to provide high user friendliness. If a person wants to give his data to someone, he shows his visit card. In the same way, a card represents a user's partial digital identity, therefore someone can have several cards with different attributes. These cards can be shown to different resources for the purpose of authentication and can be saved either on a server or locally.

CardSpace follows the user-centric identity principle; the user has full control over the system and over his attributes.

The way in which CardSpace functions is presented in the next graphic:

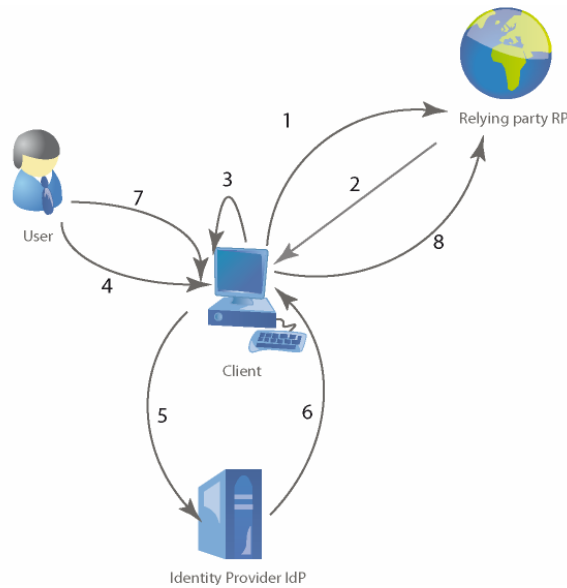


Fig. 4-3 CardSpace functionality (CardSpace 2008)

1. The client wants to access a resource of the relying parts and sends a request to this instance.
2. The relying party supports CardSpace and sends a request for the identity of the user in form of a card. In this step, the necessary *claims* are also transmitted.
3. The identity selector makes a pre-selection of the existing cards based on the conditions of the relying party. The cards that do not comply with the request are not visible to the user.
4. The user chooses one of the cards, according to his wishes. The main criteria here are the particular *claims* that he wants to transmit. If desired, the user can add more optional attributes.
5. The client sends a request for the chosen card to the identity provider.
6. The identity provider sends the desired card back to the client. The card now contains all the *claims*, which were previously certified by the identity provider.
7. The user receives the card and can verify the attributes.
8. The card is submitted to the relying party.

(Nanda 2006; Microsoft 2007)

As CardSpace is a proprietary solution for which there is no source code at the moment, it is not possible to use it to implement biometric authentication. Nevertheless, this solution has some interesting concepts and, according to Microsoft, will be able to work together with other existing AAI's.

#### **4.4.5 Sxip**

This AAI was created by Dick Hardt, one of the creators of the user-centric identity management and of the *Identity 2.0* concept, and it is designed upon the URL-based identity, where an URL is used instead of the username for the authentication. According to Dick Hardt, the protocol is no longer being developed, but the project will be adjusted to the OpenID 2.0 standard. (Hardt 2008)

#### **4.4.6 OpenID**

##### **4.4.6.1 Concepts of OpenID**

The last investigated AAI, which was finally used for the practical implementation of the project, is OpenID. The protocol that is used by OpenID was developed by Brad Fitzpatrick, the founder of LiveJournal, and it belongs to the category of central ticket systems. OpenID is based on the principle of URL identity, which towards regular usernames has the advantage that URLs are a distinct identifier. Additionally, the user has the possibility to reveal in his URL some personal information about himself, in form of a home page or a digital visit card.

The design of OpenID consists of three parts:

- Identity Provider: The IdP stores the user's credentials and other attributes that the user can explicitly share with OpenID applications (service provider). Its mission is to implement the login process in order to provide the user with an online ticket which he can later show to the service provider.

Important in this context is that the actual login procedure is not specified. This is performed by all current implementations of OpenID by means of user name / password combinations, but can just as well be realized using biometric authentication. Specified is merely the response of the identity provider or the ticket, which contains information about the success or failure of authentication.

- Service Provider: The service provider is the resource that the user wants to get access to. For this, he has to input his OpenID URL in a special input form. Then, he is redirected to his identity provider, who provides him with a ticket that the service provider can verify.

- URI: The Uniform Resource Identifier is the URL that serves as username. It is often stored on the same server as the identity provider. An example for a URI is <http://matthias.uni-regensburg.de>, [uni-regensburg.de](http://uni-regensburg.de) being the identity provider.

The URI can be also stored on a different server, e.g. <http://matthias.older.es>. In this case, the head area of the HTML document on the home page of [matthias.older.es](http://matthias.older.es) has to contain a standard link to the URL <http://matthias.uni-regensburg.de>. This link is parsed by the service provider that redirects the user to <http://matthias.uni-regensburg.de> for login.

The advantage of a division of URI and identity provider URL is the fact that the user can take e.g. the URL of his own home page as identifier. On the one hand, he can give information about himself; on the other hand, is less dependent on the identity provider. If the user decides to change from [uni-regensburg.de](http://uni-regensburg.de) to [pip.verisignlabs.com](http://pip.verisignlabs.com), he can still use <http://matthias.older.es> as his URI and only has to change the link to his identity provider.

#### 4.4.6.2 How OpenID works

The way in which OpenID functions is explained in the following graphic:

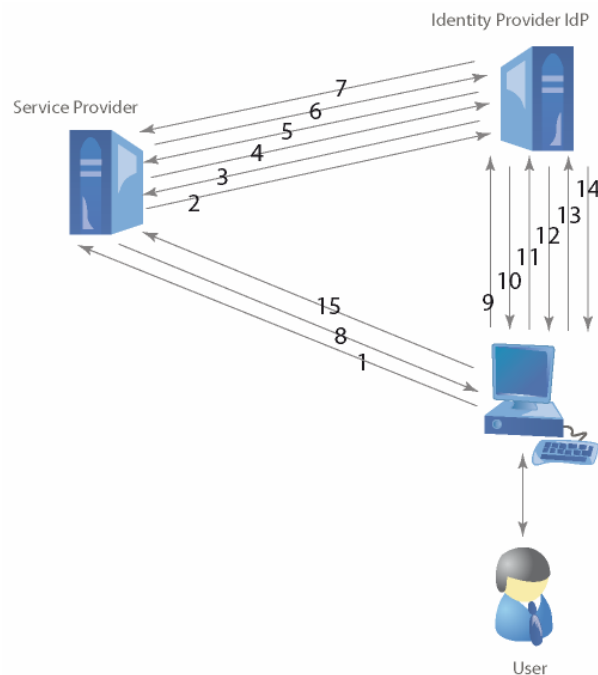


Fig. 4-4 How OpenID works

1. The user opens the web site of the consumer (service provider) and types his identity URL, for example <http://matthias.uni-regensburg.de>
2. The service provider opens the URL that was introduced by the user in order to receive the identity page. It is not mandatory that the identity page should be managed by the identity provider that stores the identity of the user. The user can store his identity page anywhere on the web and it is therefore possible for the user to choose his own URL. This URL we will call claimed identifier.
3. The service provider receives the identity page of the user. This document specifies the address of the XRDS document, which can be managed by the IdP, not by the user. This document is an XML file which is used to describe metadata about a web resource.
4. The service provider requests the XRDS document from the IdP.
5. The IdP delivers the XRDS, which describes all the extensions that this IdP supports.
6. Now, the consumer and the IdP set up an association based on a shared secret and a signature algorithm (SHA1 or SHA256). The shared secret is issued by the identity provider and is usually encrypted with a Diffie Hellman shared secret.
7. The server sends back the shared secret and his public part of the Diffie Hellman algorithm.
- 8 and 9: The user agent is redirected to the IdP.
- 10: The user opens the login page.
11. He inputs his login data in the form of a password.
12. The IdP presents a set of attributes that were claimed by the consumer.
13. The user has the possibility to determine which of his attributes he wants to share. Depending of the case, some of them may be required by the consumer, others may be optional. Should he decide not to share an attribute marked as required, the service provider can deny him the access.
- 14 and 15. The IdP generates a signature of the chosen data using as key the secret share and sends this together with the data to the consumer, who checks this signature by means of the received data and the shared secret. Should the two signatures be identical, the data was not manipulated on the way.

#### 4.4.6.3 New features of OpenID 2.0

The following section compares the innovations of OpenID 2.0 as opposed to OpenID 1.1.

##### 4.4.6.3.1 Better extensions support

The most important extension in the OpenID 1.1 protocol is the Simple Registration Extension used to send attributes from an identity provider to a consumer. For this purpose, a consumer can send the following a request to the identity provider:

```
Openid.sreg.required=nickname
```

This means that the sending of the nickname attribute to the consumer is mandatory. In order to process this request, the identity provider needs to know in advance that “openid.sreg” implies the Simple Registration Extension.

Problems can occur when a special Psylock extension is defined by the name “openid.psy”. It is possible that there is another extension with the same name. Therefore, the name “openid.psy” is not explicit for the identity provider and he does not know which extension is meant by the abbreviation “psy”.

OpenID 2.0 solves this problem by using so-called Uniform Resource Identifiers (URI) already known from XML. A request conform to OpenID 2.0 could contain the following statements:

```
Openid.ns.psy=http://psylock.de/ext/1.0  
Openid.psy.required=nickname, lastlogin, lastsample
```

With the stated URI <http://psylock.de/ext/1.0>, the extension “psy” is explicitly defined and the identity provider can transmit the requested attributes if it supports the Psylock extension. The URI can also contain information about the extension used.

##### 4.4.6.3.2 Large requests and replies

OpenID 1.1 uses HTTP redirects in order to exchange data between identity providers and service providers. Therefore, the length of a message is restricted by the maximum URL length supported by standard browsers. Internet Explorer is the limiting factor as its limit lies with 2.083 characters. This limitation does not cause problems with simple authentication processes; however, the limit can easily be reached if large amounts of data are to be transferred with this extension.

Since version 2.0, it is possible to send messages as HTTP post and so eliminate the former restriction. This is technically solved by sending a form with data in hidden fields to the users' browser. A JavaScript on-load handler sends the data to the other party.

#### 4.4.6.3.3 Directed Identity

A further functionality mainly related to the discovery process is a directed identity. With this option, the user does not directly state the URL identifying him as a user (e.g. matthias.myopenid.com) to the consumer, but only the URL of his identity provider, in this case myopenid.com. Hereupon, he is forwarded by the consumer to the stated identity provider where he authenticates with his user name and password or with biometrics. Not until the following authentication response does the consumer get to know under which user name the user is known to the identity provider.

This makes it possible for the user to be registered at one identity provider but to collaborate with various consumers under different user names. Therefore he can dynamically define which identifier is to be sent to a consumer after the authentication at an identity provider. This effect can be used like a one-time E-Mail address: instead of the usual identifier matthias.myopenid.com, the user can take the identifier 12345.matthias.openid.com for another consumer. An advantage of this option is the protection of user privacy: the user can prevent web providers from creating a surfing profile by using different identifiers.

Another effect is an easier login process for the user: if he logs in to his identity provider and a previously saved association with a consumer exists, he only has to enter his identity provider (myopenid.com) at the consumer instead of entering matthias.myopenid.com as before. As he is already logged in at the identity provider, he can now log in to the consumer automatically.

If a consumer works only with a limited number of identity providers, the login process can also consist of a list of identity providers from which the user chooses the IdP he wants to log in to instead of entering it manually.

The technical implementation of directed identity is very simple: instead of the identifier matthias.myopenid.com, a standardised URI is sent to the consumer. (OpenID Specifications 2008) On the basis of this URI, the consumer can detect that it is a direct identity. Upon authentication at the identity provider, the consumer can see in the *id\_res response*, under which user name the user is known. The following HTML discovery site is parsed by the consumer and shows him that a direct identity should be used:

```
<head>
<link rel="openid2.provider openid.server"
href="http://www.myopenid.com/openid/server.bml"/>
<link rel="openid2.local_id openid.delegate"
href=" http://specs.openid.net/auth/2.0/identifier_select"/>
</head>
```

#### **4.4.6.3.4 Provider Authentication Policy Extension (PAPE)**

As the login process is outsourced by the consumer to the identity provider by the use of OpenID, it is important for the relying party to know in which way the user authenticated to the IdP. Furthermore, it can specify that the authentication be carried out only according to a certain safety standard.

PAPE enables the consumer and the IdP to agree upon specific policies marked by URIs. Common policies can be found on (PAPE policies 2008).

The three most common PAPE policies are:

- Phishing resistant: An authentication safe against phishing (e.g. Cardspace).
- Multi-factor: the user authenticates by multiple factors, e.g. by knowledge-based, possession-based or biometric attributes.
- Physical multi-factor: as previous with the addition that one authentication factor must be physical, e.g. smartcards.

#### **4.4.6.4 OpenID as implementation platform**

OpenID can be used as an AAI platform for biometric authentication due to several reasons.

For once, the user-centric identity management concept is implemented in an exemplary way. The user has the choice between many available identity providers where he can store his data. If necessary, he can even set up his own identity provider. Additionally, the user has the possibility to explicitly determine which attributes will be sent to consumers.

Then, OpenID is not limited to preconfigured trust networks, but can be used together with any standard conform identity or service provider in the web. This is one of the reasons which led to the fast expansion of OpenID in the web; in 2007 there were about 4500 consumers and over 120 million OpenID accounts (OpenID 2007). Important examples are AOL, which offers its own IdP at [opened.aol.com](http://opened.aol.com), or Sun Microsystems, which uses OpenID in its intranet. Future developments

of OpenID include a Firefox browser plug-in, which will make the data and attribute management easier. Microsoft has announced that it will support OpenID (Microsoft 2007) in the Vista operating systems in combination with CardSpace. This rapid growth of the system predicts a good future expansion of the system.

Another plus is the fact that there are implementations and frameworks of OpenID in almost all web programming languages; the implementation is well-documented and easy to accomplish even for small websites.

Ultimately, OpenID is a very good platform for biometric integration purposes because it is user friendly, easy to implement and meets the highest security requirements.

## *Chapter 5*

### **5 BIOMETRIC AAIS**

---

Adding biometrics to AAI systems leads to more security but also implies more complex system architectures that suffer from different biometric problems. For example, biometric methods present the problem of replay attacks, which can also manifest itself at the level of biometric AAIs. This chapter presents the possible architectures and elaborates scenarios that show the importance of biometric problems for AAI systems.

---

#### **5.1 Authentication methods in AAIs**

In order to increase the security of the authentication process, two different approaches can be used:

1. Stronger password: this variant uses a single password as before, but it conditions that the password should be longer than a certain number of letters, with big and small letters, special characters, etc. This method does provide better protection against brute force attacks, but at the same time it is more difficult for the user to remember, which may force him to write the password down, thus creating a higher security risk. Increasing the password length does not bring any additional protection against serious threats such as key loggers (KeyloggerPro 2008). Evidently, this approach has reached its limits and has to be replaced or extended.
2. Multi-factor authentication: this method combines the knowledge-based factor (password) with a possession-based factor (token, smart card) or a being-factor (biometrics) (Ying-Han 2007). The more factors are used, the more secure the authentication will be. This approach is more flexible than the previous one and therefore followed in this work.

For the authentication of the AAI a two-factor authentication is proposed: password and typing cadence biometrics. Biometrics have the feature of being in possession of the user all the time, they

cannot be lost or given away, they can hardly be stolen and they give the only authentication mechanism that can bind a username to a certain person.

The AAI presented uses the typing cadence biometrics based on the Psylock method of recognition (Bartmann 2004), developed at the University of Regensburg since 1993. This biometric method of typing cadence recognition uses as input parameters the pressed or released key events, together with the time when these events occurred in milliseconds.

This biometric method has the advantage of not requiring extra sensors except of a standard keyboard, is less vulnerable to key logging attacks and provides good person recognition.

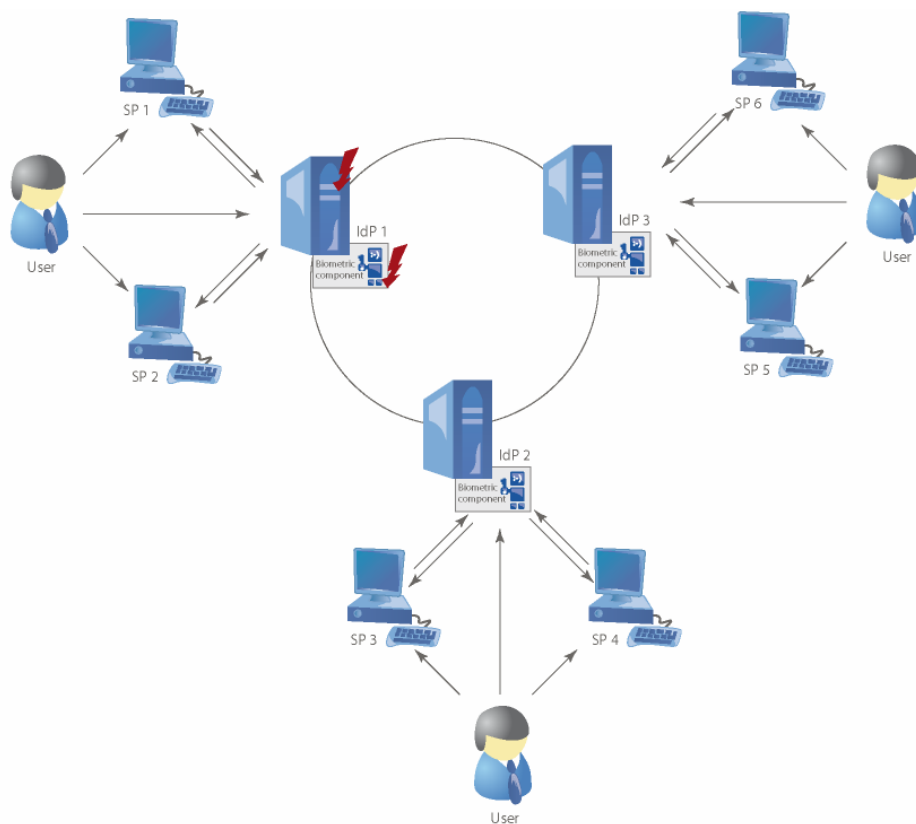


Fig. 5-1 Biometric authentication in a circle of trust requires changes in both IdP and biometric component

In order to combine AAI with biometrics, changes have to be made on both parts: the AAIs must support the additional attributes that the biometric method needs and their authentication modules have to be extended; while biometrics must meet the high security standards required by the consumers and IdPs. These changes are presented briefly in this chapter and more in-depth later in this work.

## 5.2 Architectural models

Upon adding biometric components to the AAI, three new biometric AAI architectures have to be considered:

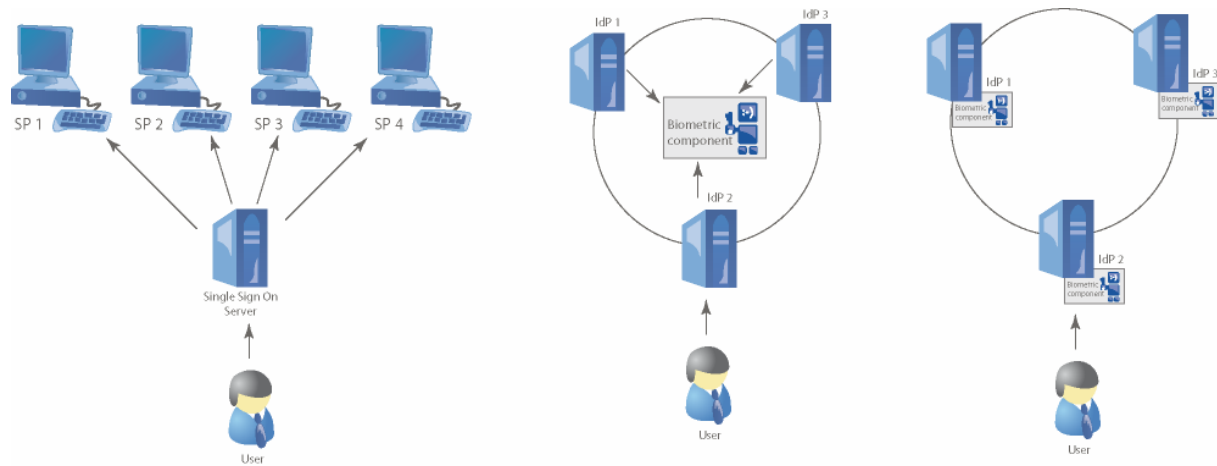


Fig. 5-2 Biometric AAI architectures

Central Single Sign On with biometric authentication:

Extending the SSO server to biometric authentication is one of the solutions that involve the least effort, as it assumes the mapping of only one server to one biometric method.

Federated AAI with central biometric component:

This architecture is practically a combination between federated AAI and central SSO server. In this case, the different IdPs within the federation divide one single biometric component that is responsible for verifying the authenticity of the person that accesses the online resources. Additionally to the “single point of failure” problem, it is also very difficult to map the different users from different IdPs to only one user in the central biometric component.

Federated AAI with divided biometric component:

This architecture assumes that each IdP has its own user data together with its own biometric component. This architecture is the most conform to the principles of the Circle of Trust and therefore was used in this work’s model.

Two other possible cases have been left out from considerations and may be addressed in future research:

AAI with client-stored biometric component:

This architecture assumes that the biometric component is stored on the client side. In this case, the user has, on one hand, the full control over his biometric data (ideal case), on the other side, he has to take care that this data is not lost or corrupted. This architecture is also highly vulnerable to replay attacks and therefore was not considered in this work.

Combinations of SSO and circle of trust or of several circles of trust:

This example is possible when a company with more subsidiaries using a Single Sign On joins a group of similar companies in a circle of trust. This model is very similar to the one used in this research, the results achieved can also be used for the model presented.

### **5.3 Problems of biometrics that influence the biometric AAI**

Biometric methods differ from password based authentication, as they need a longer training phase (enrolment), they change with the time and cannot be replaced if they are lost. These particularities must be kept in mind while designing biometric AAI. There are three main factors of influence that these systems obey:

Architectural aspects:

In order to design the architecture of a biometric AAI, it is necessary to take into consideration the fact that all biometric features of a person are aging. The solution to this is a built-in intelligent adaption mechanism that is aware of these changes or, in some cases, a function for sharing biometric profiles. The standard function that biometrics provide against aging is template adaption, which is a process that takes place every time when a new biometric sample is put in the system. In case of biometric AAI, the template adaption must be made with consideration for the fact that different identity providers may share older biometric data.

Quality aspects:

The quality of the results that biometrics provide depends of the sensor (feature recorder) that acquires the data and of the initial training process (enrolment). In case of a federation, all the IdPs have to have the same quality requirements for the biometric methods supported.

Security problems:

Same as passwords, biometric methods are vulnerable to replay attacks. Combined with AAIs, this risk increases due to the fact that it is more attractive for an impostor to get access to many online services at once.

### 5.3.1 Replay attack as a problem for AAI systems

In order to understand these problems and their consequences for biometric AAI systems, the following use case is considered: a federation with several IdPs, all of which support biometrics as an authentication method.

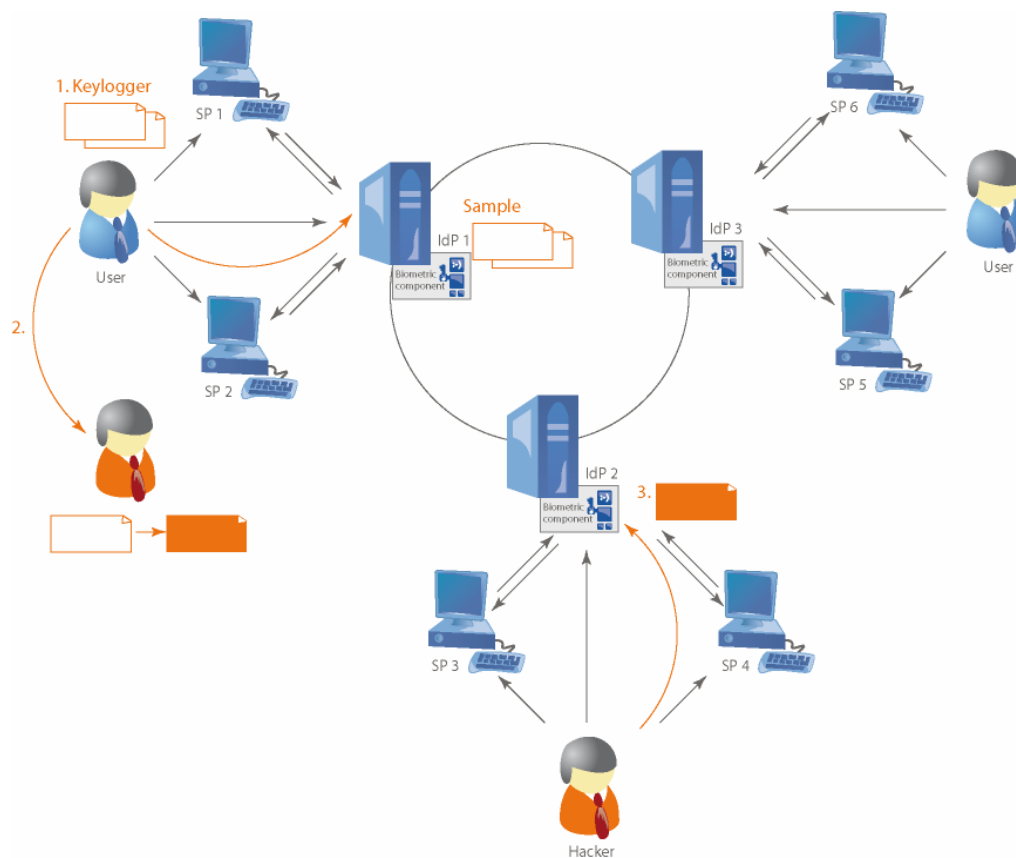


Fig. 5-3 Replay in biometric AAIs

In a federated AAI with a shared biometric component there are several IdPs, each one with its own biometric data. The user Alice logs in to IdP1 using a password and typing cadence biometrics. It is possible for the attacker Bob to install a key logger on Alice's computer, thus recording both URI, password and a typing sample *S* that belongs to Alice. While it is possible that Bob manipulates the sample *S* by changing the key order or the times (and creates a new typing sample

S' that resembles very much the original S), it is impossible for Bob to replay this new sample at the IdP1, as the biometric component will correctly recognize that the new sample S' resembles too much the sample S which is already stored in the database. Nevertheless, it is possible that Bob goes to another IdP from the circle of trust, where he can successfully replay this sample.

It is therefore necessary for the AAI to have a protection mechanism against such attacks. This mechanism has to respect the following rules:

- Not to contradict with the other factors that influence biometric AAIs;
- The changes in the logic flow of the circle of trust have to be reduced to a minimum;
- If possible, no changes for the Service Providers, only at the level of IdPs;
- If possible, to resolve other problems, such as aging of biometric data.

The questions that have to be answered are: How can replay attacks within an AAI be recognised? Should the biometric data be synchronised? Are there any other ways of solving this problem? An in-depth analysis is made in the chapters 10 and 11, where the basis of two biometric AAI prototypes is presented.

### **5.3.2 Quality of biometric data as a problem for biometric AAIs**

The same scenario can be also applied for the case of the quality of biometric data. If Alice needs more biometric sensors (for example, because she is using several computers, each equipped with a different device), then she will have to create several biometric profiles. In our case, Alice authenticates by means of typing behaviour and has two profiles ("standard pc" and "notebook") stored at IdP1. Alice has a biometric profile at another IdP from the circle of trust, where she made the enrolment procedure for only one profile ("standard pc"). If she has to authenticate at IdP2 with her notebook, the use of a different sensor will create recognition problems in the authentication process, due to the different quality of the biometric data.

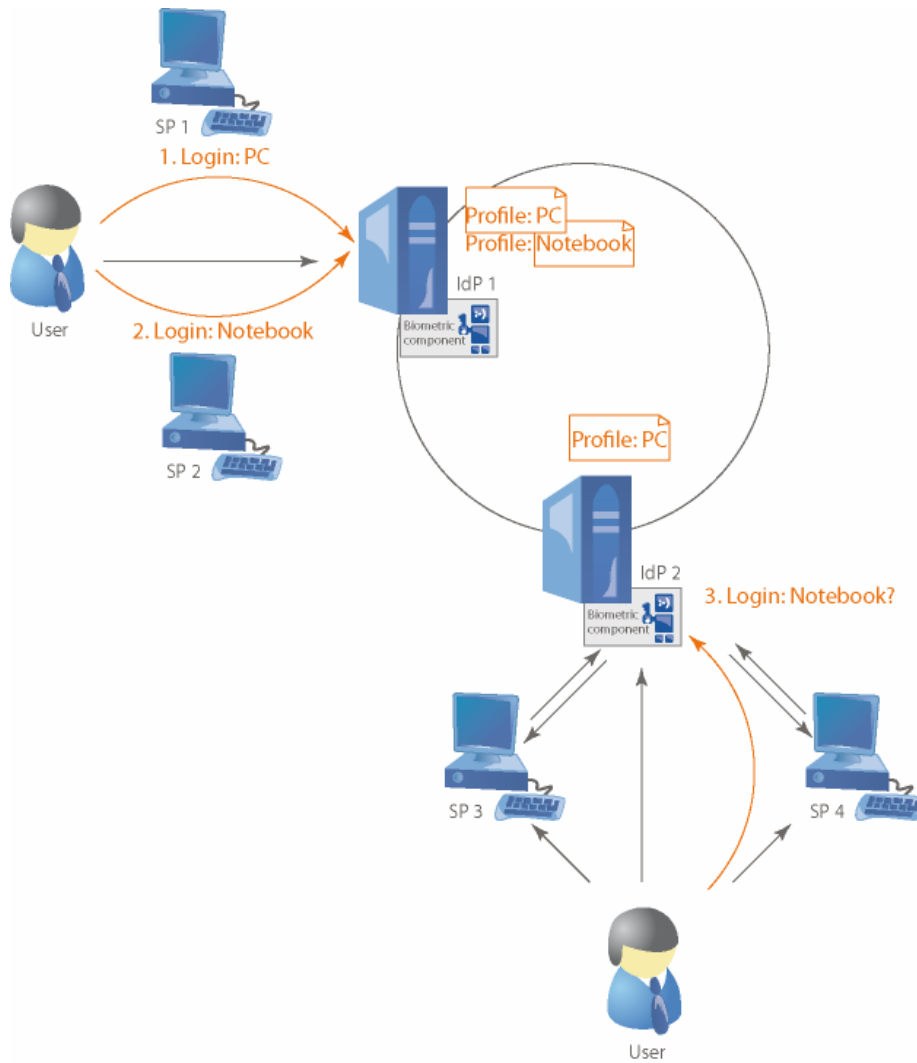


Fig. 5-4 Quality problems in biometric AAIs

In this case, it is necessary to investigate the way in which the profiles can be shared or the fact that quality specifications like threshold or the type of profile should be considered as attributes and shared between IdPs.

This problem is also treated at the level of typing behaviour biometrics in chapter 7 of this work.

### 5.3.3 Aging of biometric data as a problem for biometric AAIs

In our system configuration, where Alice can log in to several IdPs in the same circle of trust, it is possible that she has preferences, which means that she logs in more often to some IdPs (like IdP1) than to other IdPs. While the biometric data stored by IdP1 is recent and the template is regularly adapted, the data saved by IdP2 is aging. After a period of time, Alice will be rejected upon logging

in to this IdP, as her biometric features have evolved too much and differ from the pattern which is stored on that server.

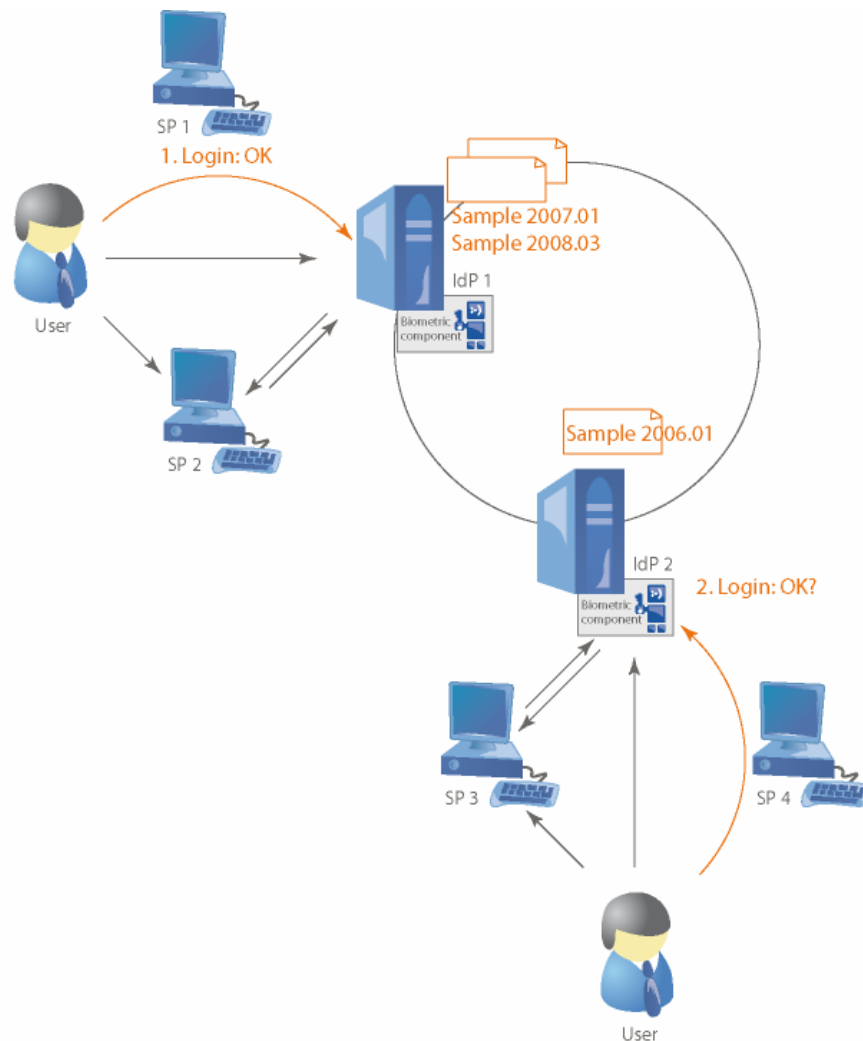


Fig. 5-5 Aging in biometric AAIs

In order to solve this problem, an investigation at the level of the biometric method must be carried out in order to determine the time in which the biometric data is aging. Is it an option for IdP1 to share the newer biometric samples with other IdPs, in order to actualize the profile at IdP2? What other mechanisms can be used to prevent this problem?

## 5.4 Conclusion

The investigations in order to provide answers for these questions have been conducted at two levels:

- The in-depth level of the biometrics in use (typing behaviour): to investigate the technical aspects of these problems and to simulate them outside of the AAI structures, in order to observe and gain more information about them.
- The general level of the AAIs: the conception of an architecture which is resistant to these problems or mechanisms to correct them.

The main two procedures which this work will investigate are the following:

#### 1. Synchronisation of biometric data

This can be made either at the database level or at the circle-of-trust level. By synchronisation, the biometric data will be actualized between all IdPs. This raises several issues such as the quantity of data that has to be transferred, the fact that some IdPs may be offline during the synchronisation plus problems when user names are different, or simply due to the fact that current protocols do not support real time synchronisation upon login. This procedure makes sense only for closed networks, where several IdPs assure a redundant authentication.

#### 2. Remote authentication

Considering the fact that the user has only one biometric identity, he has to have only one IdP which stores his biometric data. For a federated environment, only one identity is necessary to access all the resources within the circle. The other IdPs cannot make an authentication of the user, as they do not have the required authentication data.

The proposition on hand is a *remote authentication*, which implies that when the user tries to use an IdP from the circle of trust where his biometric data is not stored, that IdP will contact the “home” IdP of the user (in our example IdP1) and simulate the behaviour of a consumer (service provider). If the IdP1 confirms the identity of the user (thus storing his biometric data and checking it for replay attacks), the other IdPs can also allow him to enter the circle of trust.

The next chapters will treat these problems at the level of biometrics and of AAIs and practical solutions will be offered.

### **6 REPLAY ATTACKS IN BIOMETRIC SYSTEMS BASED ON TYPING BEHAVIOUR**

---

In the previous chapter, the different problems of biometric AAs were presented. It has been shown how replay attacks manifest at the level of AAs. Now it is interesting to zoom into this problem on the example of typing behaviour biometrics. Is it possible to create an algorithm that can recognize such attacks?

---

#### **6.1 Security problems in IT-systems**

Nowadays, the rapid growth of information technology and the continuous development of new other technologies make IT-security play a very important role. The trend of using the internet channel for both private and business transactions from e-business and e-commerce makes it necessary to empower the IT-systems with proper protection mechanisms. (Eckert 2008)

In order to minimize or fully prevent errors, these protection mechanisms follow well-defined security goals including the authenticity, the integrity and the confidentiality of information. These goals must be achieved in order to have a secure system. Nevertheless, this proves to be very difficult and complex in practice, as these goals compete with each other; in order to increase the authenticity of data, one must make compromises as to its integrity or confidentiality. Therefore, it is obvious that 100% security can never be achieved. In this case, it is necessary to make a prioritisation of security goals and by that to make compromises in the practical implementation of these goals.

An analysis can provide a measurement of the possible threats that may occur, which gives information about the possible security leaks or finds the places where an attack would cause the most damage and that requires particular protection. This technique of analysis can be used not only to develop a security concept, but also by a possible attacker in order to determine the weak points of a system.

There are two types of attacks: passive and active ones. The passive attack consists in the illegal gaining of information about protected data sources or systems, therefore passive attacks aim on the security goal of confidentiality. The second way of attack is active, where the intruder also modifies or even destroys the data he has gained access to; therefore it aims for the goals of data integrity and availability. (Eckert 2008) Examples for passive attacks are monitoring, sniffing and key logging, while denial of service, spoofing and phishing are examples of active attacks.

The special cases of man in the middle attacks and replay attacks are combinations of both passive and active threats.

## 6.2 Security problems of biometric systems

The fast progress of biometric authentication has brought with it not only advantages in quality, comfort and security, but also the necessity of consideration for several biometrics specific problems.

A security problem that should not be disregarded is the error rates. Due to technology limitations and other factors (e.g. statistical and heuristic calculations), biometric systems cannot provide 100% recognition, but always require a threshold. The use of such thresholds brings two kinds of errors, which can influence the security of a system differently. For once, an unauthorised user can be falsely recognized as the real user and therefore access to the system being granted. In order to prevent this, the equal error rate EER has to be kept as low as possible. (Eckert 2008)

Beside the error rates problem, biometric systems are also susceptible to other risks. A list of these is presented in the following table:

Type of attack	Examples	Possible counter measures
Spoofing and mimicry attacks	Artificial finger used on fingerprint biometric device	Multimodal biometrics, vitality detection, interactive authentication
Fake template risk	Fake template stored on the server	Encryption, intrusion detection, smart cards
Transmission risk	Data intercepted during transmission, during enrolment or data acquisition	Interactive authentication, rejection of identical values, system integration
Cross-system risks	The same template used in a different application with different security levels	Hash functions, encoding algorithms
Component alternation risk	Malicious code, Trojans, etc.	System integration, well-implemented security policy
Enrolment, administration and system use risk	Data altered during enrolment, administration or systems use	Well-implemented security policy

Type of attack	Examples	Possible counter measures
Noise and power loss risks	Flashing light to optical sensor, changing temperature or humidity of fingerprint	Well-implemented security policy
Power and timing analysis risk	Power analysis and differential power analysis garner data on biometric template	Noise generator, low power consumption chips in biometric devices
Residual characteristic risk	Fingerprint remaining on the sensor copied by various means	Technology assessment, multimodal access
Similar template, similar characteristic risk	An illegitimate user has a template similar to the legitimate user	Technology assessment, multimodal access, calibration review
Brute force attack	An intruder user brute-force to deceive the system	Account lock after a number of unsuccessful attempts
Injection risk	Captured digital signal injected into authentication system	Secure transmission, heat sensor activated scanner, date-time stamps in digital representation of images
User's rejection	The invasive nature of biometric techniques could cause users to reject using the system	Training and awareness of users and the selection of the least intrusive technique possible
Changes in physical characteristics	Some techniques depend on face or hand characteristics, but these human aspects change with the time	Monitoring of features, template adaption
Costs of integration with other legacy systems	Coherence with other techniques used for legacy systems than have to be integrated	Cost-benefit analysis
Loss of data	Hardware failure	Data backup and restoration

Table 6-1 Risks of biometric systems and countermeasures (ISACA Group 2008)

From the point of view of security, one of the most important threats is replay attacks, which will be investigated for case of typing behaviour biometrics in this chapter.

### 6.3 Replay attacks

The replay attack is a form of threat which manifests in the repeated sending of data recorded previously. From this definition it is evident that a replay is a similar form of the “man in the middle” attack. A replay has a passive and an active component. The passive component shows in the fact that a data communication is recorded. The active component consists in the re-sending of information acquired from the sent data packages.

This threat has a tendency of increasing in time. The explanation lies in the fact that encryption algorithms and authentication methods have become more and more complex and secure, so that breaking this security, for example by means of decryption, is not possible due to the high effort on the side of the attacker. With a replay attack, the hacker is not forced to break complicated systems anymore, he just has to wait long enough to record and then replay the input data to access those systems.

Another reason for the relevance of replay attacks is the broad spectrum of application areas where such an attack is possible. Beside the field of normal authentication, where a username and a password could be recorded, another new field emerged, the biometric systems. Replay attacks are more dangerous at the level of biometrics, as once the biometric feature has been lost and is in possession of the hacker, the person cannot use this feature anymore. It is also comparably easier to get the biometric features of a person, as they are visible to everyone and leave traces everywhere (e.g. fingerprints on a glass).

### **6.3.1 Protection against replay attacks**

Due to the high danger that comes from the problem of replay attacks, it is important to investigate adequate protection mechanisms. All countermeasures start from the assumption that a hundred percent protection cannot be guaranteed. Despite that, several measures give a high level of protection. In case of authentication systems, the most important security criterion is the way in which the authentication itself is conducted, for example password or biometric. Aside from choosing a more secure way of authentication, there are some other procedures that will increase the system's protection:

- Secure encrypted communication channel: login data should not be sent in clear text to the authentication system.
- Recording of login data through sequence numbers: especially in biometric authentication systems we can provide the biometric samples with a sequence number and to determine whether a particular sample has been used before.
- Use of signatures: the login data, either in form of username and password or biometric, can be provided with a digital signature, which is a cryptographic method to confirm the origin of a data sample and by this to certify that the data is not fake.

- Physical protection mechanisms: these play an important role particularly in the case of biometric systems and are meant to protect the sensor of the biometric sample from attacks from the outside and to make sure that no other sensor can be inserted instead.

- Algorithms or replay recognition: the last and the most effective method, which is nevertheless very difficult to implement, is the design and use of algorithms which can distinguish an original sample from a replay one. This technique is also being discussed in this chapter and comes down to creating a function:

```
checkReplay(newBiometricSample, sampleCollectionFromDatabase[])
```

This has to return true when the newBiometricSample is found in the database and false if the sample is original.

## 6.4 Key logging

Beside many other variants of malware that is wide spread at the moment and that is used by hackers, one of the most dangerous variants is key loggers. These are espionage tools (in either hardware or software form) that are installed on the computer and that can record all the key inputs the user makes, thus transforming it in a serious threat for the private sphere of persons or companies. Through key loggers, hackers can also reconstruct text that was typed and receive important private data like passwords or similar credentials. They are different than other threats like viruses or worms, as they do not spread through the network, but work as standalone programs. It is nevertheless not excluded that key loggers disguise as useful applications that also record user inputs in the background (similar to the threat of Trojan horses).

Most key loggers have only the function of recording the user typing and can additionally have some extra functions like:

- The recording of running processes after a predefined schema;
- Screenshot acquiring after a fixed time plan or at the occurrence of certain events;
- Copying of clipboard contents.

The gathered information is stored on the hard disk in clear or encrypted form and then sent to the author via e-mail, web or another network protocol. (Zaytsev 2007)

#### 6.4.1 Susceptibility for replay attacks

Typing behaviour presents a more secure way of authentication than a normal password. Still, it is possible that typing behaviour is susceptible to replay attacks. The difference is that in the case of a password, once the attacker has recorded it, he can use it immediately without any problems, while in the case of typing behaviour he has to process the recorded data in order to use it again. Two cases can be distinguished:

- Fixed text: the user authenticates by biometrically typing the same text, which can be a sentence of about 50 characters like:

`"Hello User: It never rains in Southern California."`

In this case, once the attacker has recorded this sentence, he can start the process of resending the keys in the system.

- Variable text: the user authenticates by biometrically typing different sentences. In this case, the attacker must wait long enough to capture more of these sentences, so that he is able to re-generate every possible key combination a new sentence may have.

It is also possible to embed the text that the user has to type in a graphic form that is not easy to decode by a machine (Captcha 2008) and to ask the user to immediately begin typing. This method can substantially diminish the danger of replay attacks.

- Challenged text: the user types a text which is fully random and which appears on the screen as a response to what he has typed so far. This method also gives good results in stopping replay attacks.

The problem in case of typing behaviour is the fact that, while dynamic text inputs are good in preventing replay attacks, they require a longer system training (enrolment) from the side of the user and that the biometric method itself needs longer text inputs as compared to the fixed text variant (Bakdi 2007). Therefore, the fixed text variant is more popular, as it also gives the possibility of "password hardening" by increasing the security of a password with typing cadence (Biopassword 2008).

The following scenario is possible: an attacker installs a key logger on a user's computer with biometric authentication based on typing behaviour and is able to record the authentication procedure exactly in the way the user made it, with all the personal features and dynamics. In case

of modern computers, the small time delay that is necessary for the extra key logging of the hacker tool is not noticed by the user. Now the attacker is in possession of a biometric sample of the user. This sample he can even manipulate in order to change some key strokes or key events, with care that the new produced sample should still resemble the original, else the system will not recognize him. Then he can start the login procedure and, when asked to input the fixed text phrase, he can start replaying the recorded sample, which can be eventually accepted by the system.

This is only one possible way how the attacker could get into the systems by means of replay. To be more exact, there are 8 possibilities where the hacker can start his replay attack. In order to understand them, the process that happens when the user logs in biometrically has to be examined in-depth.

At first, the user must train the system in a process called enrolment. With this, he provides the system with several biometric samples. These samples are recorded with a sensor (in our case, a keyboard); they are sent from the sensor to the PC which can make some pre-processing of the raw data; then the pc submits the data to a server, which receives it and forwards it to a biometric component. This component can interpret whether the data is qualitatively good enough and store it in a database. When the biometric component decides that there are enough samples available, it creates a biometric template, which is also stored in the database. This template has all the biometric features of the user, which were extracted from the received samples. After this, the user can log in by sending another sample, which will be matched by the biometric component against the template that was previously generated. In case the score is higher than a predefined threshold, the user is granted access to the system. Depending on the biometric method, the template can be recalculated considering also the new sample acquired.

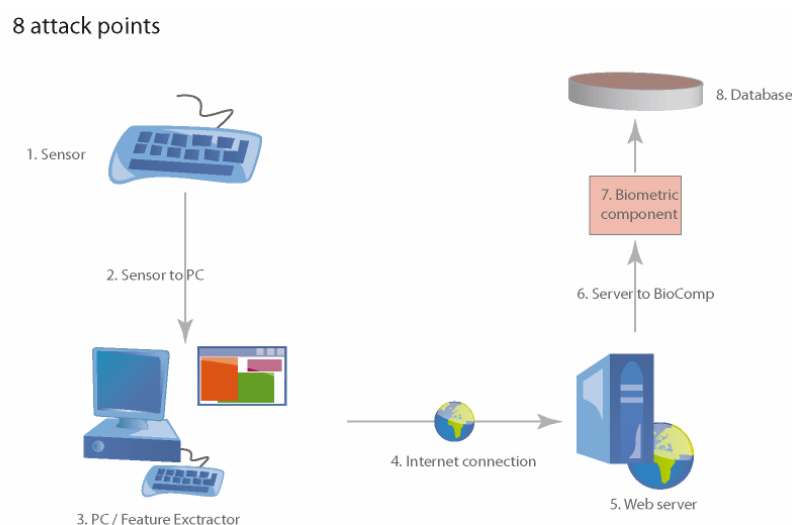


Fig. 6-1Replay attack scenarios (Ratha 2001)

The 8 possible attack forms, together with the countermeasures that can be taken are presented in the following table:

Place	Replay attack	Countermeasure
Sensor	A hardware key logger is built in inside the biometric sensor.	Make sure that only trusted persons have access to the biometric sensor;
Sensor to pc connection	A hardware key logger is attached to the keyboard sensor; the sensor has a wireless connection to the pc.	Visually check the connection between the sensor and the pc; make sure that the wireless connection is encrypted with the newest encryption algorithms.
PC (feature extractor)	Very common point of attack, usually a software is installed that records key events.	Use an antivirus and a firewall; notice if the system becomes slower during biometric authentication; use of an anti-key logger.
Internet connection	Man in the middle	Secure connection to the server
Web Server	A person has access to the server	Secure the server, limit the number of administrators that can access the biometric data
Server to biometric component	The connection is not secured	Encrypted connection, if possible not wireless.
Biometric component (decision maker)	Manipulations of the threshold	Storing of this component on a different server, higher security, minimal access even to administrators. Use of a replay algorithm to determine whether the sample is replay.
Database	A hacker can connect to the database	Storing the database in an „identity vault“ (Doupnik 2007)

Table 6-2 Replay attack attempts

All these attack scenarios have one thing in common, which is that, independently on the place where the replay attack is started, the data (whether original or replay) lands in the database. This is an important point for creating algorithms that can recognize replay samples and filter them from the originals.

## 6.5 Replay Algorithm

The possibility that an attacker can log in by means of a replay sample makes it necessary to design an algorithm that can recognize replay attacks. In this chapter, the places where such an attack can take place have already been presented. From these we can see that the biometric component is the

place where measures have to be taken at the level of the biometric method itself, everywhere else the counter methods are a common problem of IT security.

The algorithm developed in this work is called `checkReplay`. It requires two types of data as parameters:

- `newSample`: this is the recorded biometric sample on which the algorithm has to make the decision whether it is replay or not;
- `sampleCollectionFromDatabase[]`: all other samples stored in the database that belong to the authorised user.

The algorithm must comply to several conditions in order to function effectively:

- The algorithm must use a threshold to determine whether the sample has a “too high” similarity with any of the samples from the database. This statement is based in the empirical observation of (Bakdi 2007) that samples from the same user still have a measurable potential of difference and, even in case of persons with a stable typing behaviour, identical samples do not occur. The algorithm must return a replay match score. Other than at normal biometric matching, where a high match score indicates the user and a small match score the attacker, in the case of replay, higher replay match scores show that the sample was too similar to some previous data, thus pointing to a replay attack, while smaller replay match scores show that this data was not available until now, therefore it is not replay.
- The algorithm must not replace the biometric method itself, it must determine that the sample is similar or not to what is stored in the database only by means of statistics.
- The speed of this algorithm depends on the number of samples the user already has. The more samples are stored in the database, the slower the algorithm will work. This can be prevented by trying to use either replay-generated checksums of the samples, which store the sample in a form that is already prepared for the algorithm. Another possibility is to take into consideration the fact that the typing behaviour is aging, so that only the newest samples (either the last  $n$  samples or all the samples which were typed in the last  $m$  months) are considered by the algorithm.
- The influence of the algorithm on the FAR has to be high, that is the number of the false acceptance rate has to decrease. On the contrary, its influence has to be low on the FRR, which means that the user must not be additionally rejected by the replay algorithm.

- The algorithm must work correctly with any other protection method used.
- If the attacker modifies the replay sample, the algorithm must determine these changes and the impact of the changes upon the match score returned. For small changes, the impact must be high (higher replay match score), for many changes low (small replay match score). However, in case of many changes of the original sample, the replay sample will be automatically rejected by the biometric method, as it is not corresponding to the typing profile of the user.
- For security reasons, all samples that were typed under a certain username must be stored. It is possible that the user tried to authenticate, failed due to the fact that he was not attentive enough, while that sample was captured by an attacker, who can modify it to remove the mistakes of the user and send it again. If this sample was not stored in the database, an attack would be successful.
- The samples that were recognized as replay must not be deleted, but also stored in the database for future checking.
- For performance reasons, the replay checking must be done only for samples which were already recognized as belonging to the real owner by the biometric method.

The algorithm can be divided into several phases, which are presented as follows:

a. Receiving the samples from the database:

At the beginning, the biometric component establishes a connection to the database where the user samples are being stored and reads either the samples themselves or their checksums. The result will be stored in an array called `sampleCollectionFromDatabase[]`.

b. Receiving the sample to be checked:

In this step, the sample that has to be checked is received from the server and converted into a format that is accepted by the algorithm. This format has to be compatible with the format of the samples from the database or their calculated checksums.

c. Comparison of two or more samples:

This step is the core process of the algorithm. In this step, the new sample will be tested against every other sample which was received from the database. In case of a positive match (too high similarity with one or a combination of more samples), the process will stop. Otherwise, it will process all the samples and generate for all replay match scores against the new sample.

d. Decision:

If the greatest replay match score is higher than a predefined threshold (this threshold can be user or system specific), the new sample is recognized as a replay attack, marked as such and stored in the database. The user will receive an error message. If the score is lower, the sample is recognized as original and access is granted to the user.

### 6.5.1 Core of the checkReplay function

The checkReplay function consists of several parts that have to be closely considered. As previously mentioned, this function accepts two parameters, one in form of a biometric sample, the other one as a sample collection.

In this context, a sample is defined as a string with a predefined form, formatted at the level of the feature extractor on the client side. An example of such string is presented below:

```
2008-01_01_13:10:53&066v0000&065v0031&066^0016&065^0047&067
v0156&067^0031
```

2008-01-1\_13:10:53 - Date and time when the sample was originally typed;

„&“ - Begin of a new event;

First 3 digits - ASCII key code of the pressed or released key;

„v“ or „^“ - shows whether the key was pressed („v“) or released („^“);

Last 4 digits give the time in milliseconds that has passed since the last event.

The previously shown example shows that the following keys were pressed:

b ↓ a ↓ b ↑ a ↑ c ↓ c ↑

with the corresponding times:

b ↓ a ↓ = 31 milliseconds

a ↓ b ↑ = 16 milliseconds

b ↑ a ↑ = 47 milliseconds

a ↑ c ↓ = 156 milliseconds

c ↓ c ↑ = 31 milliseconds

For reasons of simplicity, we consider that there is only one sample in the database, that is:

```
sampleCollectionFromDatabase[] = oldSample
```

and

```
checkReplay(newSample, oldSample).
```

The question at hand is whether the new sample is a replay or not, that is whether there is an old sample in the database which resembles the new sample more than a certain threshold.

For this, we extract from the two sample strings the keys, key up and key down events and the corresponding times.

Note: An interesting resemblance comparison can be made at the string level, using a function like Levenstein (Levenstein 2008), which would return the number of differences between the two strings. However, the moment when the attacker changes the order of some events in the string, this function will immediately return very big differences between the two strings.

The next step is based on an empirical observation which shows that, on Windows NT operating systems (2000, 2003, XP, Vista), upon pressing keys, the time when these keys are registered by the operating system is a multiple of 15, with one or two milliseconds extra noise. These problems of raster and noise will be discussed in the chapter about quality of biometric data. For the moment, it is important to know that in the next step we remove this noise using the following function:

```
NewTime (ms) = int(OldTime(ms) / 15) * 15
```

The times from the previous example will also be rounded like in the following example:

```
b ↓ a ↓ = 30 milliseconds  
a ↓ b ↑ = 15 milliseconds  
b ↑ a ↑ = 45 milliseconds  
a ↑ c ↓ = 150 milliseconds  
c ↓ c ↑ = 30 milliseconds
```

This procedure is necessary as the operating system inserts this noise of 0-3 milliseconds at every key event, thus manipulating even the replay attack and making it really difficult for the algorithm to compare even exact events.

Note: This noise removal is not made at the level of the biometric method, as there the noise is important for the mathematical calculations that allow user recognition.

After this, the next step is to put these events in a 3 dimensional array with the following elements:

- X axis: all key down and key up events;
- Y axis: all key down and key up events;
- Z axis: all repeated events (for example, the combination b ↓ a ↓ can occur more times in a sentence.

In our example, the array has only two dimensions (no double key events) and has the following structure:

	a ↓	b ↓	c ↓	... ↓	a ↑	b ↑	c ↑	... ↑
a ↓		30						
b ↓								
c ↓					150			
... ↓								
a ↑						45		
b ↑	15							
c ↑			30					
... ↑								

Fig. 6-2 Array generated from a sample.

This procedure is made for both samples, old and new.

Note: It is also possible to include here the order in which the keys have occurred, by adding the events in the 3<sup>rd</sup> dimension (not displayed here). In the experiments made at the University of Regensburg it has been shown that the algorithm gives better results when this order is ignored.

After this process is finished, the two samples are brought in a format which is now easy to compare. The two arrays are parsed and it is calculated how similar these two arrays are. This procedure is shown below in pseudo code:

```

Begin getSimilarityPercent
  for (x=0 until Maximum X): Iterate over X-Axis
    for (y = 0 until Maximum Y): Iterate over Y-Axis
      if (MatrixA [x] [y] == MatrixB [x] [y])
        Increase the number of similar events with 1
        Increase the number of total events with 1
      else
        Increase the number of total events with 1
      end for
    end for
  end for
  ReplayMatchRate = Similar events / Total events * 100
  return ReplayMatchRate
End getSimilarityPercent

```

Based on this replay match rate and a certain threshold it can be determined whether the new sample is too similar to the old one; that is whether the new sample is a replay sample for the old one. The value of this threshold will be measured later in this chapter, based on exact test results.

For overview, here is the logic flow of the replay algorithm:

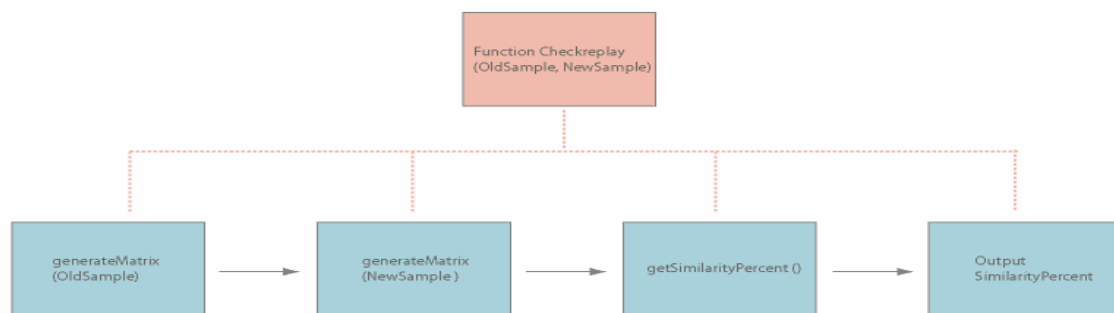


Fig. 6-3 Logic flow of the replay algorithm

In order to check the functionality of this algorithm, we must make a set of empirical tests with normal and replay samples and see whether the algorithm can distinguish these.

### 6.5.2 Test environment

Before describing the actual test phase it is important to understand the environment in which the test was made, as it explains the basis of this work. The test system was a PC with an installed Apache Server controlling the necessary web applications:

- The biometric method, providing user recognition functionality;
- The recorder application, by means of which the data was acquired by the system;
- The replay recognition component, which was using the database of the biometric system.

Additionally to that, key logger software was programmed which had the function of recording the key inputs and, if desired, modifying them or retype them.

As database was used the free version of PostgreSQL, which was filled with two types of data:

- 70000 user samples from ca. 2000 users, samples which were assumed to be replay-free.
- 500 user samples from ca. 10 users, including samples which were replay.

The test had three phases. In the first one, it was checked whether it is possible to distinguish at all between the replay and normal samples by means of a smaller quantity of data. This phase implied that it should be possible to manually detect which samples were replays.

The second phase assumed a bigger quantity of data (500 user samples), where the algorithm had to prove that the replay recognition functions correctly.

In the third phase, it was investigated the influence of the algorithm over normal user data, to see whether some of the samples that exist in our database (70000 samples) would have been recognized as replay.

### **6.5.3 Test phases**

In the first test phase it is necessary to make a “proof of concept” for this replay algorithm; that is to check whether the method would work at all or not. Therefore it is necessary to verify how similar the typing samples of a user are to each other and how similar is one original sample with several replay samples. Another thing which had to be tested in this phase was whether the noise removal process does not have influence over the user samples in the way in which they would be identical, thus making the replay recognition process impossible.

For this, several samples from the same user were gathered and plotted as follows:

- x axis: the number of the event;
- y axis: the time in which this event occurred.

As the users had to type a pre-defined text, in this phase the plot did not include the exact key code, but only the key order.

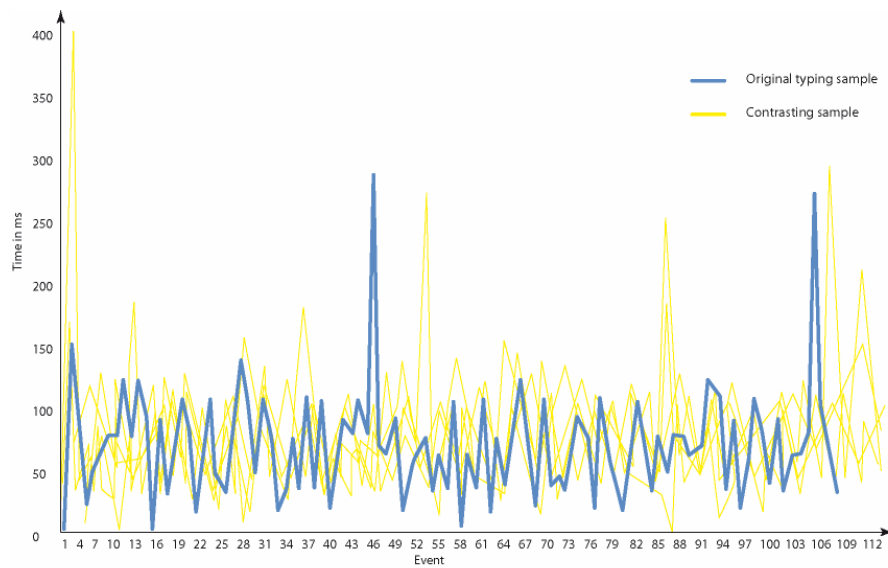


Fig. 6-4 Original vs. 5 typing samples from the same users

The next part of this test was to determine how similar a user sample is to more replay samples. For this, a fixed sentence of ca. 50 characters was typed by several users and was at the same time recorded with a key logger. Afterwards, the sample recorded by the key logger was automatically retyped 5 times.

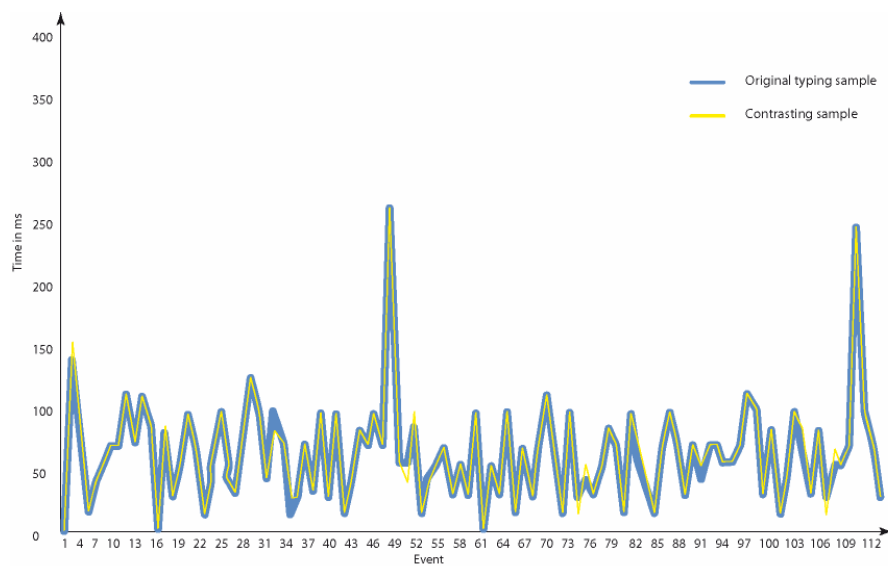


Fig. 6-5 Original vs. 5 replay samples

After the noise was removed in both plots, it is easily visible that, while there is a big similarity between the user samples, they are not identical. The user samples can be still distinguished from each other, which means that the typing behaviour of the user is strongly fluctuating (as assumed) and that it is impossible for a user to type two identical samples (this statement will be proved later in this chapter).

The second diagram shows that, upon noise removal, the replay samples are almost identical to the original user sample, having an almost identical curve.

The results of this test show that there is good potential for the implementation of the algorithm and the testing with bigger quantity of data.

The test phases can be divided into two categories: manual and automatic tests. The test which was described above belongs to the category of manual tests. The problem in this case is the fact that one can check only 2 samples. Another disadvantage is the fact that the key logger cannot make variations to the original typing sample, which makes it difficult to test the algorithm in real-life conditions.

For the second phase of the test some parameters were changed in the following configuration:

- The test has to be made automatically;
- It has to simulate the real replay-attack conditions;
- It has to work with data which was not obtained in the laboratory.

For this I have used a database filled with biometric data from ca. 10 persons that have enrolled and authenticated (altogether ca. 500 typing samples). While enrolling and authenticating, a key logger recorded the key inputs and replayed them later. The purpose of this test was to check the replay algorithm for errors; therefore, the normal and replay samples were marked in the database and later verified whether they were correctly recognized as such.

With this method FAR and FRR curves can be calculated for replay, exactly as a normal biometric method, which lets us have some information on the quality of the replay protection algorithm.

Before generating the replay FAR curves, three types of replay were defined and marked in the database to which type each sample belongs to:

- Type 0: original sample, no replay;
- Type 1: normal replay, the sample does not differ from the original one;
- Type 2: time-based replay (several times were changed in the replay sample)
- Type 3: combined replay (two original samples were combined in order to achieve one replay sample).

The samples were then sorted in the following way: for each original sample, the replay samples that belong to it were marked and selected, together with their type. The replay protection algorithm calculated replay match scores, which were in the end plotted in a graphic.

Type 1 replay:

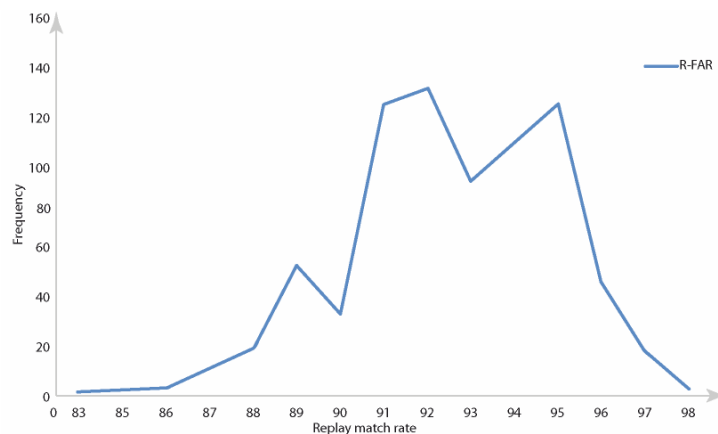


Fig. 6-6 FAR for “type 1” replay

From the FAR curve of the “type 1” replay can be determined that most match scores are in the area between 90% and 97%, therefore the algorithm can correctly recognize this form of replay attack.

Type 2 replay:

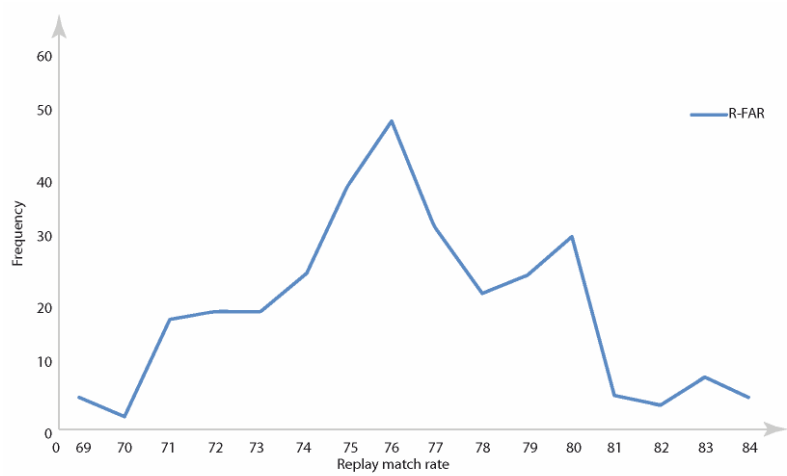


Fig. 6-7 FAR for “type 2” replay

The replay samples of “type 2” are also correctly recognized, but here the recognition rate lies between 73% and 80%.

Type 3 replay:

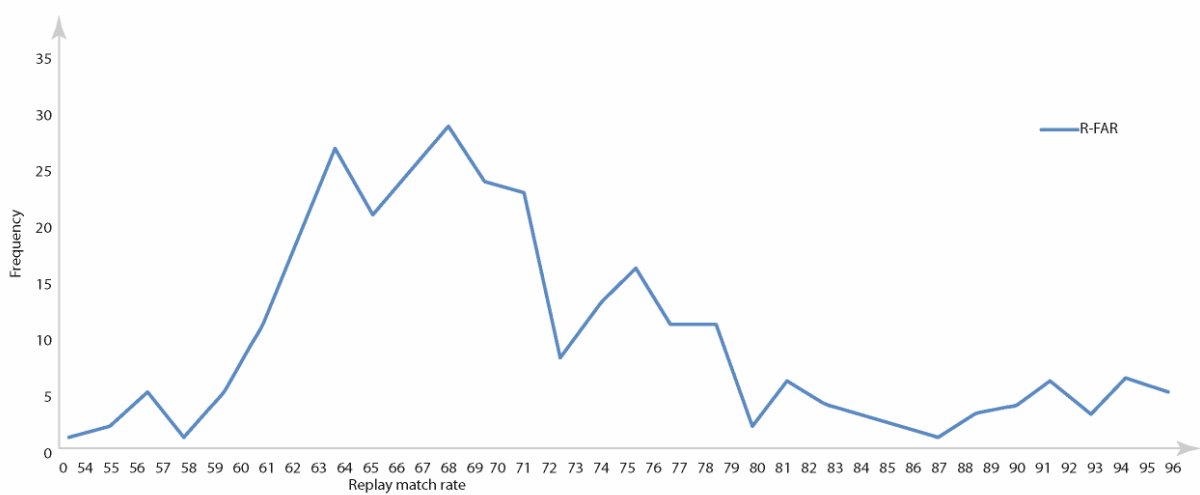


Fig. 6-8 FAR for “type 3” replay

The results of the 3<sup>rd</sup> form of replay show that even when combining two samples, the algorithm can still detect the replay samples with a precision of ca. 60%-70%.

The third test concentrates itself upon the replay FRR curve, trying to verify whether two original samples from the same user are so similar that the second sample would be considered a replay, although it is not. For this, a database with 2000 users and ca. 70000 user samples was selected. The test can be described in the following pseudo code example:

```

For each user in database
    Get all the samples from the database for that user
    Sort these samples chronologically
    For each sample1 of the user
        For each other sample2 older than sample1
            checkReplay(sample1, sample2)
    Next sample
Next user

```

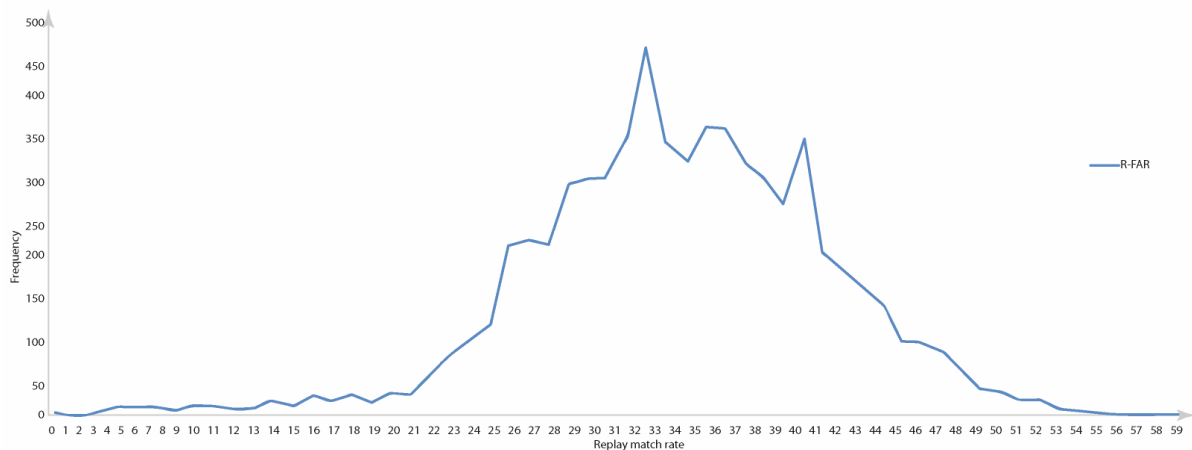


Fig. 6-9 Replay FRR for original samples (“type 0”)

From this diagram, it can be seen that between the samples of the same user there is a similarity between 26% and ca 40 %. In few cases only were there replay match scores over 50% and the maximal score was 59%.

In order to determine from which point on a sample should be considered a replay, a threshold for replay must be calculated. The threshold is the point where the replay FRR and the replay FAR meet. For this, all the three curves were merged and scaled according to the number of samples which were used to generate that curve.

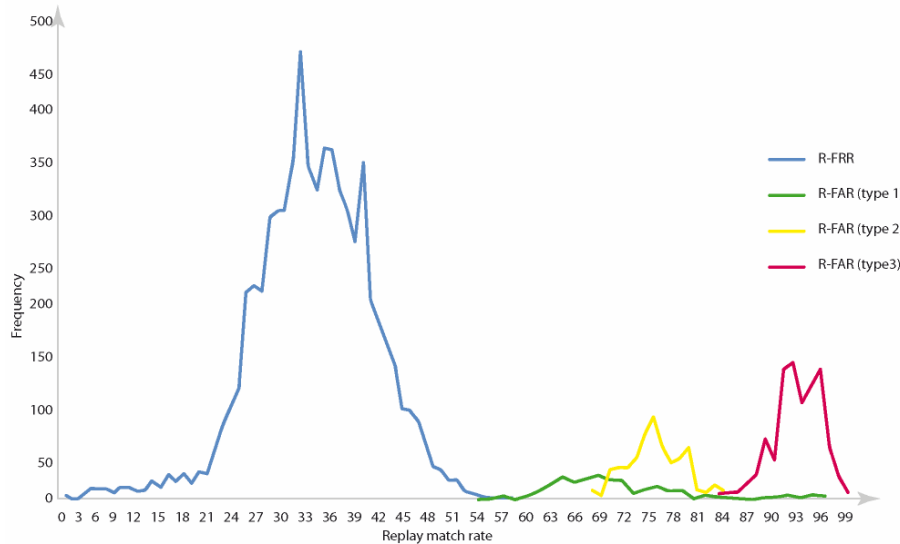


Fig. 6-10 Replay FAR and FRR curves

By means of this plot, a threshold can be set at 57%, which will not have any influence over the user samples with replay match scores lower than this level and it will offer good recognition against the three types of replay previously investigated.

Note: it is possible that an attacker modifies the sample with more times and keys or that he builds a replay sample out of more than two original user samples. In this case, the replay algorithm will probably return a match score lower than the threshold. This problem will be investigated in the next topics.

## 6.6 Extending the test procedure

The last graphic from the previous chapter shows the FRR curve of the original samples and the FAR curves of the three replay types. From this it can be seen that the replay can be very well distinguished from original samples in this test environment and that a replay threshold can be easily determined. If we consider here only the “type 2” replays, which replay a sample by means of changing the number of times, then it is clear that the quality of the replay recognition depends on the intensity of the changes, that is how strong the times between the keys are being changed. If many times in the replay sample are modified, the replay match score decreases, resulting in a change of the FAR curve to the left.

Expected trend of the FAR curve of the “type 2” replay:

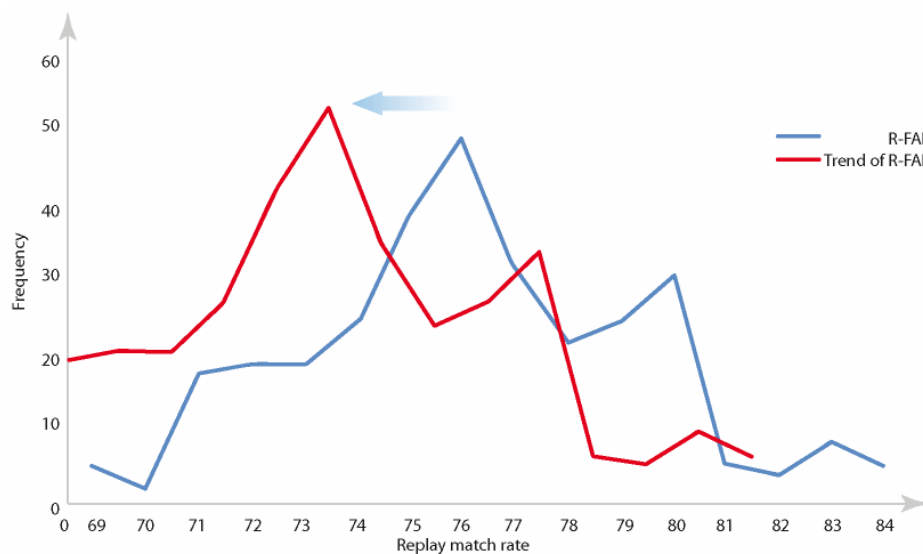


Fig. 6-11 FAR curve for “type 2” replay – trend

Through this change of the FAR curve, the number of replay samples that are falsely recognized as originals would increase (the match score would be under the threshold). In order to prevent this, we can set a new threshold that is lower and more secure. The change of threshold would have the consequence that some original typing samples with a higher similarity percent would be falsely recognized as replay and therefore the FRR of the biometrics would increase.

The same event is expected if the number of the time permutations or the intensity of these is changing or when a larger number of times changes to a higher value.

As mentioned before, the replay algorithm must not replace the biometric method; therefore if an attacker changes the original sample more than to a certain extent, the biometric method should realize that the sample does not actually belong to the real user. For an accurate replay determination is therefore necessary to take into consideration the match scores that the biometric method delivers (not to be mistaken for replay match scores).

### **6.6.1 Requirements to the new test scenario**

The analysis of the previous phase led to the result that the obtained results can be improved. For this, it is necessary to design a new test, which will consider the new requirements:

- For once, it is necessary to extend the method generating the replay samples in order to generate all possible variations of the original sample and to determine whether the algorithm recognizes them as replay or not.
- Moreover, it is needed to interconnect the replay algorithm and the biometric method, so that only the samples that have been attested as belonging to the user are checked for replay.

This test has to give information about the quality of the checkReplay algorithm. This makes it necessary to make changes in the replay algorithm, in the process of the replay sample generation as well as in the whole test procedure itself.

### **6.6.2 Extending the generation process of the replay sample**

One of the most important parts of the redesign process is the process of extending the complexity of the replay samples. The generation of replay samples has to be bounced in the direction of “type 2” replays, which are the most complex form of replay attack. For this, two new parameters have to be considered, namely the number of time manipulation and the intensity of these. In order to get a maximum number of possible replay samples from an original, the algorithm has to start from an original sample and deliver all possible combinations of replay samples. This algorithm must be kept flexible and should accept input parameters like:

- The number of time manipulations: from 0 (“type 1” replay) until all times from the sample;
- Standard deviation of the times: how much the times should vary from the original time;

Upon running the algorithm we must decide for one of the two possible methods:

- Method 1: the generation of a replay sample with a fixed number of time variations;
- Method 2: the generation of all possible replay samples.

Four scenarios for the replay samples result from combining the two parameters with the two methods:

- Replay sample with a fixed number of time and intensity changes;
- Replay sample with a variable number of time changes and fixed intensity;
- Replay sample with a fixed number of time changes and variable intensity;
- Replay sample with variable number of time changes and variable intensity.

The two methods used to generate replay samples are presented here in pseudo code.

Method 1:

Begin Method1

```

Var OriginalSample = input Originalsample
Var Count = Number of time changes
Remove time stamp of the original sample
Create arrays for [Key codes] [Up or down events] [Times]
Iterator i = 0
While (i < Count)
    Change random time in array
    i++
End While
Convert the arrays to New replay sample
Return New replay sample

```

End Method1

Method 2:

Begin Methode2

```

Var OriginalSample = input Originalsample
Remove time stamp of the original sample
Var Count = Number of events in the original sample
Var String [] array2
Create array1 for [Key codes] [Up or down events] [Times]

For (i = 0 until Count)
    While (a = 0 <= i)
        Change times in array1
        change array1 back into New replay sample s
        Insert s into array2 [i]
    End While
    i++
End For
Return array2

```

End Method2

Beside method 1 and 2, the algorithm uses another function which determines a variable value for the time changing. This function is relevant only for the scenarios 3 and 4 of the replay sample generator. In the first two scenarios, a fixed value is used which is stored in a variable that is responsible for the time changing.

The function calculating a variable time value is presented here:

```
Begin createTime
    Var lowerBorder
    Var upperBorder
    Var oldTime
    Var newTime
    newTime = random number between oldTime - lowerBorder and
oldTime + upperBorder
    or
    newTime = random number between lowerBorder and upperBorder
    return newTime
End createTime
```

By means of this method it is possible to generate the entire spectrum of permutations starting from an original sample. These permuted replay samples can be marked in the database as such, thus making easier a later analysis.

### **6.6.3 Including the match rate of the biometric system as additional feature**

In the previous paragraphs, it was mentioned that the replay match score alone is not enough to consider the quality of the replay algorithm. For this we need to include a second parameter in the test phase, which is the match score returned by the biometric method itself. The biometric match rate shows by means of statistical and heuristic methods how similar the actual typing sample is to the profile of a user; it does not compare two normal samples.

In the analysis of the test phase, the two match scores must be interpreted the following way:

- An original typing sample is the one that shows a high biometric match score and a low replay score;
- A true replay sample has either a high biometric match score and a high replay score, or a medium biometric score and a high replay score.

#### 6.6.4 Connecting the replay algorithm to the biometric API

In order to test the new qualitative measures of the replay attack, they have to be integrated it in the logic of the biometric method. For this, we consider the following case:

- Two databases, one with the source data, which must contain replay samples, and another one where the results are stored;
- $N$  samples are extracted from a user (chronologically sorted) and sent to the enrolment procedure of the biometric method in form of a collection  $S_{1-n}$ ;
- The next sample  $S_{n+1}$  is used to authenticate the user to the system;
- The sample  $S_{n+1}$  is taken and sent to a replay generator. For this, no key logger is used; we assume that the attacker has replayed the exact copy of the original sample (the noise is equal to zero);
- The replay generator starts a process which creates variations from the replay sample using the previously defined parameters;
- The replay samples are checked for biometric authentication (is the user still recognized?) and replay (can the algorithm still recognize the replay sample?)
- The data is stored in the destination database.

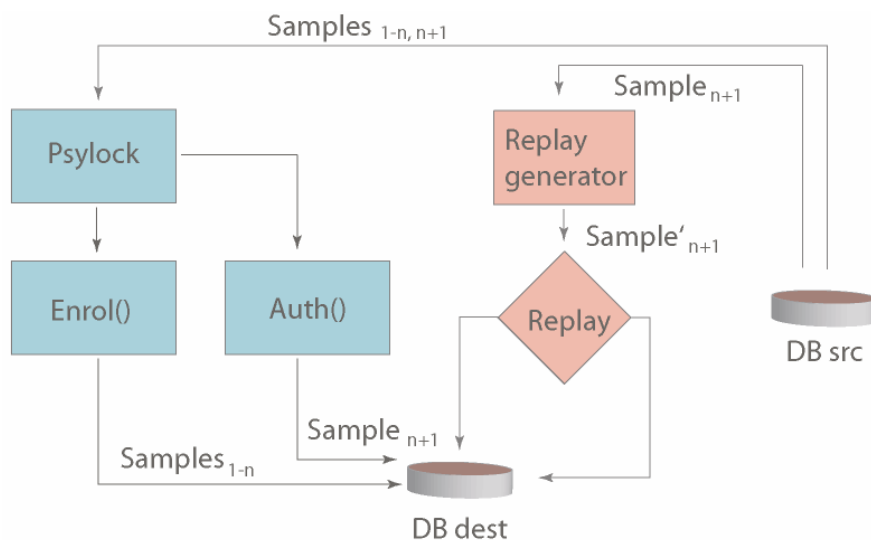


Fig. 6-12 Connecting the replay algorithm to the biometric API

At the end of this process, a data pair consisting of a replay match score and a biometric match score will be stored in the destination database *Dbdest*. This data can be used afterwards for quality analysis.

#### 6.6.5 New test results

The redesign of the replay check algorithm in correspondence with the biometric match score leads to a higher efficiency both of the biometric method and of the replay algorithm. These two parameters can be now put together in a diagram where the X axis represents the replay match score in percent (0-99,99%) and the Y axis marks the biometric method match score (also 0-99,99%). As the test data is known, we can also mark on the chart whether the results show an original or a replay sample. In case of replay samples, we can even mark which type they belong to.

In this case, the algorithm is effective when:

- Replay samples show a small biometric match score and a high replay score;
- Original samples show a high biometric match score and a low replay score.

The following cases show bad results:

- Replay samples show a small match score and also a small replay score (not dangerous for system security, as the decision is still made by biometrics – it may be the user who typed badly);
- Original samples show a small biometric match score and a high replay score (not dangerous for system security, but influencing the user FRR);
- Replay samples show a high biometric match score and a small replay score (worst case scenario).

In order to visualise these statements we can use the following data, which was arbitrarily chosen in order to give a better overview.

- For original samples: the columns marked bold show the ideal case (high biometric recognition, low replay score); all others indicate different problems.

Original Sample Number	1	2	3	4	5	6	7	8
Psylock match score	99	97	12	15	<b>97</b>	<b>89</b>	12	9
Replay match score	89	88	89	78	<b>12</b>	<b>8</b>	5	10

Table 6-3 Replay and biometric match score for original samples

- For replay samples: the columns marked bold show the desired result, which consists of replay samples that have a low biometric match score and a high replay score.

Replay Sample Number	1	2	3	4	5	6	7	8
Psylock match score	96	92	<b>8</b>	<b>12</b>	91	94	10	6
Replay match score	88	90	<b>81</b>	<b>79</b>	11	7	6	12

Table 6-4 Replay and biometric match score for replay samples

For original samples, the cases presented before can be illustrated in the following graph:

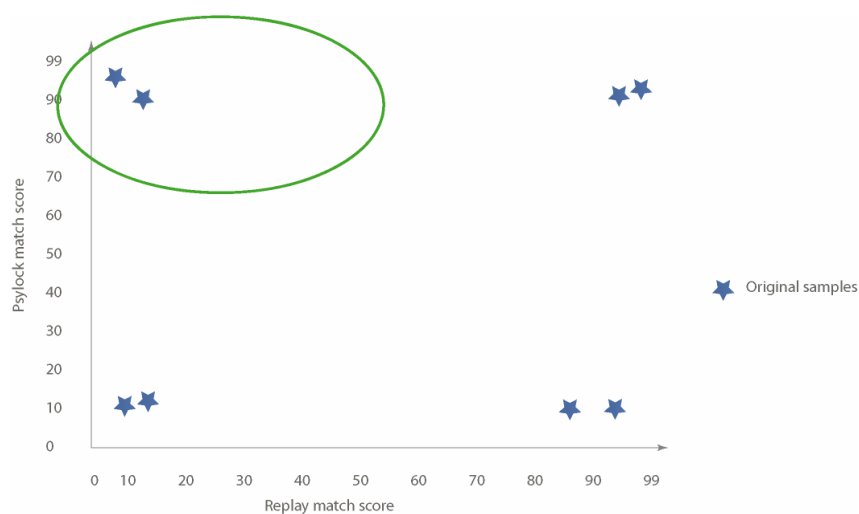


Fig. 6-13 Replay and biometric match score for original samples

In the same way, the graph below shows the possible cases of replay:

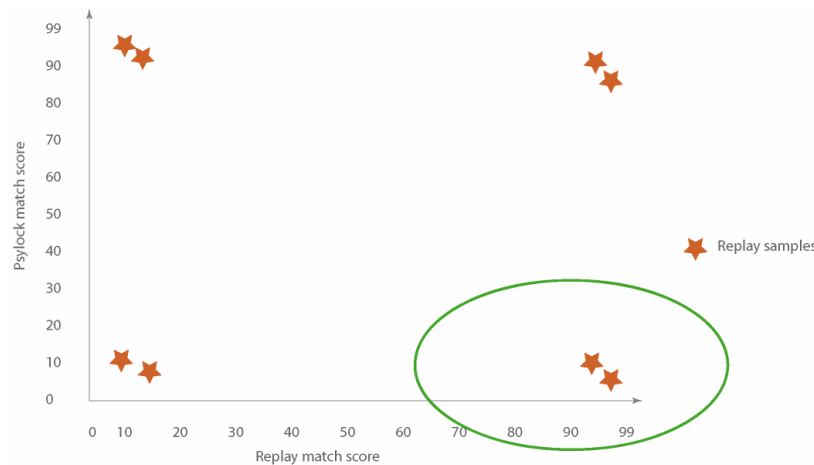


Fig. 6-14 Replay and biometric match score for replay samples

In both plots, the points marked with a circle show the ideal case in which both biometric and replay algorithm return the best results.

## 6.7 Conclusion

Investigating the problems of replay attacks at the level of biometrics can be made only using experimental data. The methods presented here can be used as a starting point to investigate replay attacks at the level of typing behaviour biometrics and offer a good protection against replay attacks.

Another possible way to optimize this algorithm is the use of the “Needleman-Wunsch” algorithm (Needleman 1970) which is used in chemistry to compare DNA sequences. This algorithm can be modified in order to compare in the same way typing samples. An advantage of this algorithm is the better consideration of the order of key events, which is made here only with the Levenstein function.

## **7 QUALITY TESTS FOR BIOMETRIC SYSTEMS**

---

The quality provided by biometric systems depends in the first line on the sensor used for data recording. Other components like browser or operating system add extra “noise” to the measurements; this may lead either to the fact that the biometric method will not work on some system configurations or worse, that the biometric template will be damaged by samples recorded in low quality. In order to prevent this, the specific quality problems have been investigated by means of empiric tests and solutions for the typing behaviour biometrics have been designed.

---

### **7.1 Quality problems of biometric systems**

As mentioned in chapter 5, the quality of biometric systems depends on the initial recording process. We distinguish between hardware problems, such as the use of different sensors, and software problems, such as problems that occur while using different browsers.

From a general point of view involving all biometric methods, the occurring quality problems are due to the big variety of sensors, each of them with different (mostly also unspecified) quality features. Another problem that has to be put into direct relation with quality is the fact that a large number of companies that produce biometric systems does not follow the general guidelines for biometrics (BioAPI 2008) but decide to use non-standard solutions for their products.

As quality requirements differ for various biometrics, we investigate the quality problems on the example of typing cadence. Here we distinguish between two types of problems:

#### **1. Software problems**

Typing cadence uses a browser to record key inputs, so the main problems are produced by variations in browsers, as there is no 100% compatibility between different browsers.

A first software problem is the fact that no browser is capable of recording the actual keys that were pressed (“a”, “b”, “c”) but only their corresponding key codes (“65”, “66”, “67”). While most of the keys are correctly recognized by all browsers (mostly keys from the left side of the keyboard), some special characters (like “period”, “comma”, “Shift”, “Ctrl” and so on) are assigned different key codes in different browsers. As typing behaviour puts a great emphasis on this kind of keys especially (for example, it is considered a typical feature for a user to type “left Shift” or “right Shift”), a bad recognition of these keys leads to a higher EER for this biometric method.

Connected to the key code problem is the issue of languages different than English, whose key codes are not correctly recognized. For languages using syllable- or word-based typing (like Chinese), the operating system blends in a special window where the user can press a couple of Latin letters and the system will automatically suggest the proper word, which the user will finally select by means of arrow keys. For the process of measuring typing behaviour, this means that, while the user presses several keys on the keyboard, the browser receives only the final word that the operating system “injects” in the browser.

Another problem is the fact that in order to read the key codes, the browser itself must receive this information from the operating system, which reads it from the keyboard driver that, in turn, receives it from the keyboard itself. This path leads to time delays that influence the measurement quality. While the path from keyboard to operating system is the same for all browsers, the browsers use different mechanisms to read this data, which leads to the fact that some browsers are “faster” (smaller delays, better quality) and some of them “slower” (big delays, low quality) in reading key codes.

The operating system itself does not read key events continuously, but only with a small delay, which results in the fact that key events will be raster to this delay. It is also possible that the system will measure zero milliseconds between key presses, which is practically impossible. Additionally, the system may insert some “time noise” which brings more delays in the measurements.

Another limitation of the operating system is the fact that measurements are made in an interval of milliseconds. For typing cadence, a more precise measurement on the side of the operating system (for example in micro- or nanoseconds) would result in a better EER.

## 2. Hardware problems

The hardware used by typing behaviour biometrics is the keyboard. This hardware can have quality problems determined by three factors.

For once, there is no universal keyboard. Each model has different particularities in shape, position or role of the key.

Then, there are several keyboard types, with variations starting with standard keyboards and finishing with notebook (more compact, smaller key size) or ergonomic keyboards, which have the keys oriented in different ways for an easier use. This leads to the fact that users have the tendency to prefer one of these keyboard types and tend to be slower in using other keyboards.

Thirdly, the different ways of transmitting information from the key itself into the system also play an important role; cord keyboards (USB, PS2) react differently than wireless models (with or without encrypted connection). On the wireless keyboards it can happen that, when pressing several keys at once, the sensor buffers the information, waits for some milliseconds and then sends all information at once to the receiver.

### 7.2 Recording key events with typing behaviour biometrics

A big advantage of typing cadence is the fact that the required sensors (keyboard) are already present in most computer systems. It is possible to record key inputs with a browser supporting one of the following technologies:

- ActiveX: recording keys using ActiveX technology is possible only in Internet Explorer, where for security reasons this plug-in is automatically deactivated. In order to reactivate it, the user must lower the security level of the browser; this is not a desired action, as it could compromise the entire system.
- Java: key events can be recorded with Java. Within an applet, key strokes can be captured and sent to a server. The advantage of Java is its support of multiple browsers and operating systems. The problem of using Java applets to record keys lies in the fact that it requires the Java Virtual Machine (JVM), which is not installed at all workstations, e.g. in an internet cafe.
- JavaScript: using this technology to record key strokes is one of the easiest variants, as JavaScript is available in almost every browser, where it is activated by default. The major disadvantage of JavaScript is the fact that there are many versions of it and each has differences in recording key

codes. The specifications of recognized key codes change even from one browser version to the other, so the use of JavaScript is recommended only in some special cases, where the computer is not allowed to use any other technology (possibly due to security reasons).

- Adobe Flash: Flash is one of the most stable technologies to be used for keystroke recording. The Flash plug-in is usually installed in the browser and all browsers use the same Flash version. The key code specification in Flash is very stable and only few key codes are not properly recognized. For these reasons, Flash has been selected for the experiments conducted in this work.

### **7.3 Software problems**

The software problems of typing cadence are in fact browser or operating system specific problems, therefore it is necessary to investigate them by means of adequate tests for different OS-browser platforms. For this purpose two components called “client recorders” have been designed to register key inputs and to send them to a server. The difference between them lies only in the technology used to design them, one recorder is programmed in JavaScript and the other is a Flash version. The purpose of these tests is to determine which version delivers a higher quality of data for biometric recognition.

For this, following tests were made:

- Raster test
- Key code recognition test
- Speed-delay test
- Foreign language compatibility
- Enrolment-authentication analysis

The tests were made under the operating systems Windows XP, Knoppix Linux and Mac OS 10. The tests and the corresponding pairs of browser and operating system are presented below. For some of the browsers, it was important to make all the necessary tests on all operating systems, while for other browsers it was enough to make the research on a single operating system. Tests like raster or key code recognition have a high dependency on the operating system, while other tests like foreign language it is not expected to receive different results in varying OS.

Browser \ Flash Tests	Raster	Speed-delay	Key code recognition	Foreign language	Enrolment-authentication
Win IE 6 and 7	○	○	○	○	○
Win Firefox 1.5, 2.0	○	○	○	○	○
Win Opera 9	○	○	○		
Linux Konqueror 3.5	○		○		
MAC Safari	○		○		
MAC Firefox 1.5	○		○		

Table 7-1 Tests with browser-OS combinations

### 7.3.1 Raster tests

“Raster” is the interval of time in which the operating system reads key events from the keyboard driver. In order to determine this raster, several browsers have been tested under three operating systems (Windows, Mac and Linux). The test consisted in a fast random sequence of key events and the analysis of the times registered in the two client recorders.

The result was a plot with the observed times against the relative number of occurrences of different times.

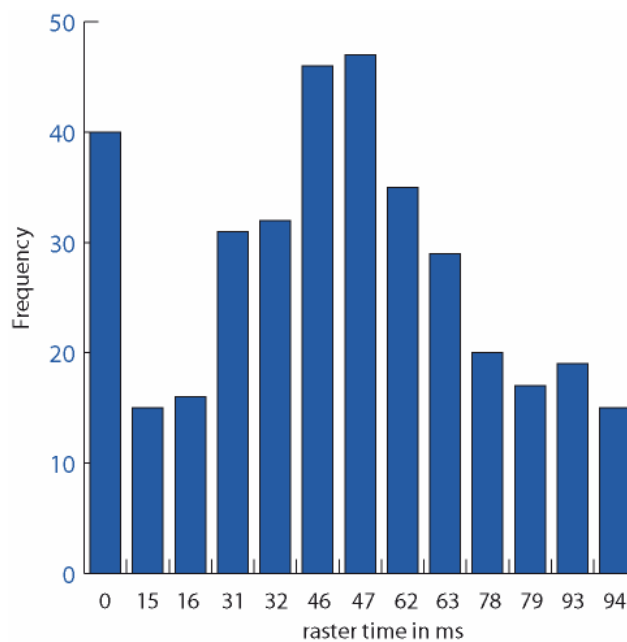


Fig. 7-1 Resolution tests under Windows

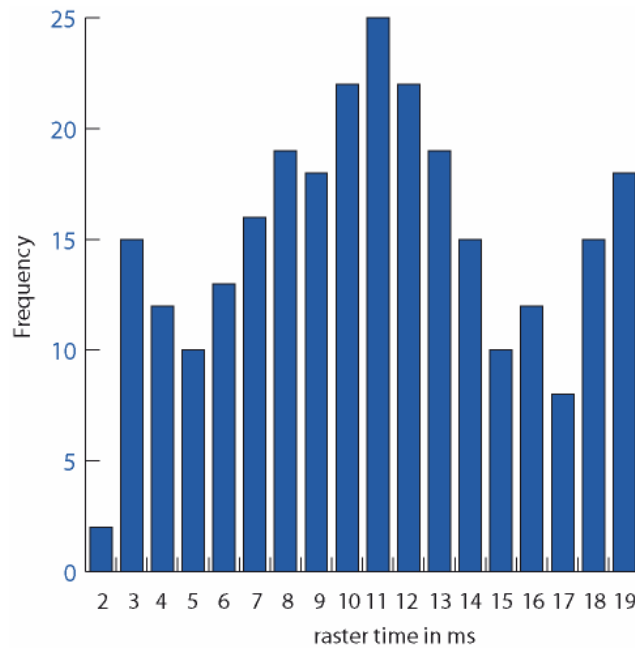


Fig. 7-2 Resolution tests under LINUX

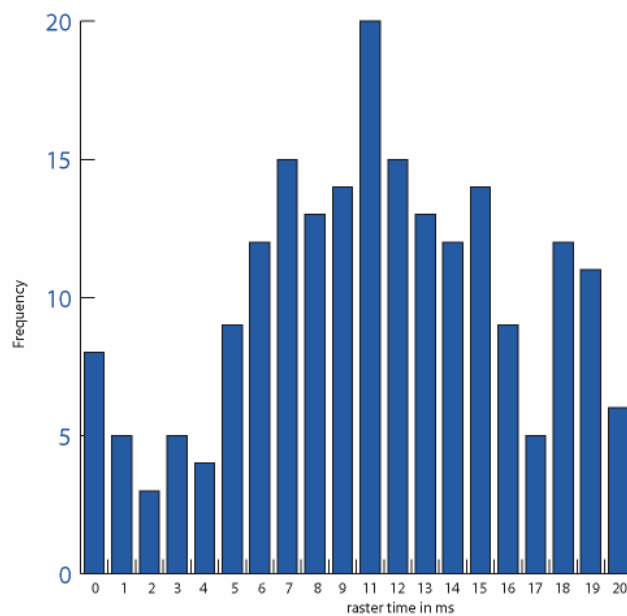


Fig. 7-3 Resolution tests under MAC

Test results: the analysis of time frames shows that a raster pattern is only visible for the Windows operating system. This raster is of 15 milliseconds. The fluctuation with 1 ms is due to the noise produced by the operating system. Under the other operating systems (Linux and Mac), there is a superior raster of only 1 millisecond, whereas Linux registered only times starting from 2 milliseconds and Mac OS also registered error times of zero milliseconds.

Note: Another possibility to repeat this test is to use a hardware sensor installed directly in the keyboard, which would give more precise measurements of the real times when the keys were pressed. A third possibility is presented in this chapter under “Speed-delay tests”.

### 7.3.2 Key code recognition tests

#### 7.3.2.1 Key code recognition in Flash

In case of the key code recognition test, it was investigated which ASCII/Unicode is associated to a certain key, in order to determine possible differences between browsers and operating systems. The necessity of this test comes from the technical problem that the operating system does not deliver the actual keys to the client recorder, but only their corresponding codes.

This test consists in the pressing of each key and comparing the output key code with the original keys that were pressed. This test was made for a standard QWERTZ German keyboard. In the following table are presented only the keys which deliver other key codes than the expected ASCII code.

Keys	Windows			Mac		Linux
	Internet-Explorer	Mozilla	Opera	Safari	Mozilla	Konqueror
ü	186	186	186	219	219	252
ö	192	192	192	186	186	246
ä	222	222	222	222	222	228
ß	219	219	219	189	189	223
'	221	221	221	221	221	-
+	187	187	187	221	221	187
#	191	191	191	220	220	51
-	189	189	189	191	191	189
1	35	35	35	97	97	35
2	40	40	40	98	98	40
3	34	34	34	99	99	34
4	37	37	37	100	100	37
5	12	12	12	101	101	-
6	39	39	39	102	102	39
7	36	36	36	103	103	36
8	38	38	38	104	104	38
9	33	33	33	105	105	33
,(Del)	46	46	46	110	110	46
0	45	45	45	96	96	45
Left alt	-	-	-	-	-	18
Alt Gr	17,18	17,18	17,18	-	-	-

Table 7-2 Key code recognition in Flash

Test results: Flash delivers the same key codes for different browsers under the same operating system (as expected, as all browsers use the same Flash version installed on the OS). However, differences occur between different operating systems, where it is even possible that some keys do not deliver any key code at all.

### 7.3.2.2 Key code recognition in JavaScript

The same test was repeated also for the JavaScript version of the client recorder. In this case, the results have shown that the differences between key codes can be noticed not only between operating systems, but also between different browsers, even between different versions of the same browser. Also, the number of keys that do not return any key code is very high in Linux, therefore measuring key codes with JavaScript in Linux is not recommended.

The keys that return different key codes are presented in the following table:

Keys	Windows			Linux
	Internet-Explorer	Mozilla	Opera	Konqueror
ü	186	59	220	-
ö	192	192	214	-
ä	222	222	196	-
ß	219	219	223	222
'	221	221	180	-
+	187	61	43	61
#	191	191	35	-
,	188	188	44	188
.	190	190	46	190
-	189	109	45	-
1 (Num)	97	97	49	-
2 (Num)	98	98	50	-
3 (Num)	99	99	51	-
7 (Num)	103	103	55	-
8 (Num)	104	104	56	-
9 (Num)	105	105	57	-
,(Del)	110	110	78	-
Left shift	16	16	16	51
Right shift	16	16	16	109

Table 7-3 Key code recognition in JavaScript

It is important to note the fact that no browser could determine the correct position of the left and right Shift in JavaScript; all read both Shift keys as “left Shift”. The preference of the user for one of these keys cannot be determined and causes a loss of quality, thus increasing the EER of typing cadence.

These tests have proved that Flash provides better key code recognition, as it uses a more stable, cross-browser version. The number of keys without a correspondent key code is much smaller in Flash than in JavaScript. It is therefore necessary for the client recorder to be equipped with conversion tables in order to deliver the correct key codes to the system.

### 7.3.3 Speed-delay tests

#### 7.3.3.1 Speed-delay tests in Flash

Another possibility to test the delays of the operating system and the browser can be made by means of a key logger that sends to the system events with a known time difference. These events are then measured by the client recorder and the difference to the original time is determined.

For this, a key logger is used to input the sentence:

“Hello User: It never rains in Southern California.”

in three different browsers under the Windows XP operating system. The sentence was automatically typed 50 times in each browser. For each event, a random time delay between 0 and 600 milliseconds was generated.

The speed/delay of the browser is calculated after the following formula:

$$S_d = \frac{\sum k[i] - \sum b[i]}{n}$$

where:

k[i] = key logger input times

b[i] = browser received times

n = number of sentences

S<sub>d</sub> = speed/delay

The results of these tests are presented below:

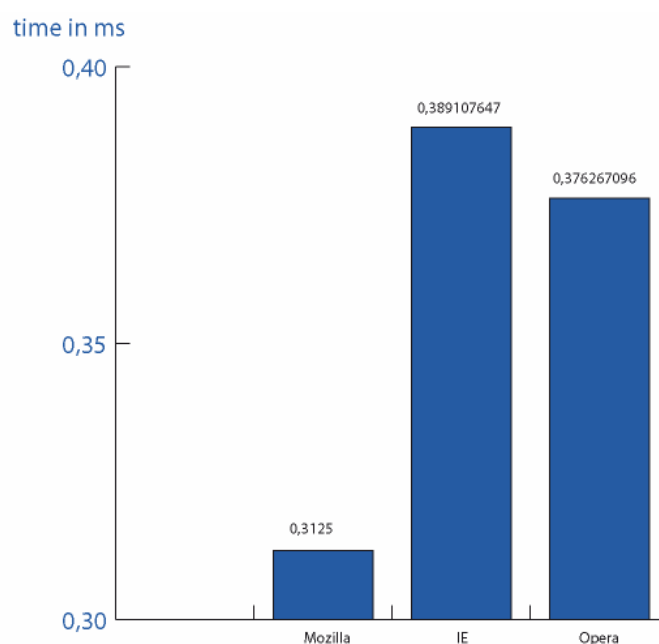


Fig. 7-4 Speed-delay in Flash for Mozilla, IE and Opera

Test result: the measured time differences between the key logger and Flash are very small (under 1 millisecond). Firefox is the fastest browser, having a delay of only 0,3 milliseconds while Internet Explorer and Opera are approximately 0,4 milliseconds slower than the key logger.

### 7.3.3.2 Speed-delay test in JavaScript

A similar test was conducted also for JavaScript. In this case, two browsers were tested (Internet Explorer and Firefox). Between these, Internet Explorer has a delay of 4,10 milliseconds and Firefox even 5,99 milliseconds.

As this delay is about 10 times smaller in Flash than in JavaScript, it is recommended to use this variant for high quality biometric data.

### 7.3.4 Foreign language compatibility

In order to see whether the client recorders can be used for word and syllable based languages, their compatibility has to be tested on the respective systems. For this test, a Chinese operating system was installed, several sentences in Chinese were typed and the output compared. For the generation of Chinese words, an input method editor has been used, which is an extra window that allows conventional keyboards to generate thousands of Chinese words by means of Latin letters.

There are two methods of typing in Chinese, one based on the phonetic pronunciation and the other on the structure of words. For each method there are several standard software solutions that can be used.

The most important method of phonetic pronunciation is called Pinyin, and it is the primary input method in China due to the fact that it is easy to learn and to use with any Latin keyboard. (Pinyin 2008) When typing in Pinyin, an additional window asks the user to type the phonetic translation of the word and, after one or two letters, the system makes some common suggestions. From these, the user can pick his choice using arrow or mouse keys, thus achieving a typing speed higher than in English or German. Approximately 90 % of Chinese typing is made using this method, but most users can type by means of other methods too.

The methods of typing after the structure of the word allow a person to write Chinese even if they do not know the language (or the phonetic writing of a word). The main product for this method is WuBiZiXing, used in approximately 15% of Chinese typing. Its advantages are the speed of typing, as for every word, maximum 4 letters are necessary, but it is more difficult to learn and requires a longer practice, therefore it is not so popular.

The language compatibility tests have been conducted with the sentence:

所有的事也是有因果循环的。

in English “whatever happens, happens for a reason”. The phonetic pronunciation of this resembles:

“suoyou de shi ye shi you yinguo xunhuan de.”

In order to type the above sentence, it is necessary to press the following keys in Pinyin:

suoyou de shi ye shi you yinguo xunhuan de。

= space key

This succession of keys appears also in the client recorder, with the difference that, at the beginning of every word, the focus jumps from the recorder window in the Chinese editor, thus that event is lost for the Flash editor. The JavaScript version does not register any key code input from the editor. Another problem that the Pinyin input method has is the fact that there are many possible phonetic translations for the same word, thus is it impossible to use a fixed text biometric method for word-based languages like Chinese.

### 7.3.5 Enrolment – authentication analysis

The most powerful form of qualitative analysis of the biometric recorder is being made by taking into consideration the results of the biometric method itself. The following test scenario is designed for this experiment:

We take a certain browser where we conduct an enrolment process by means of key logger software that has the ability of typing repeatedly in precisely the same way. The use of key loggers for enrolment and authentication gives us the possibility to repeat and compare the tests for various browsers. After the enrolment process is finished, authentication is made by means of the same key logger that simulates the same typing behaviour, with the difference that we use a different browser. The test will prove whether the browsers are compatible between each other.

For this test, we start the process of enrolment using 30 biometric samples. The authentication is conducted 10 times, first from the same browser, than from two other browsers. From this, we reach the following results:

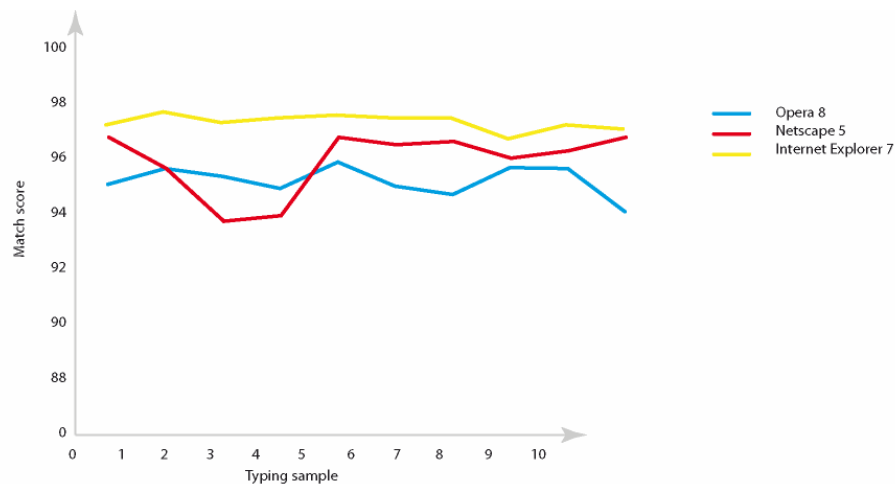


Fig. 7-5 Match scores reached by different browsers while authenticating to a biometric profile created with Opera 8

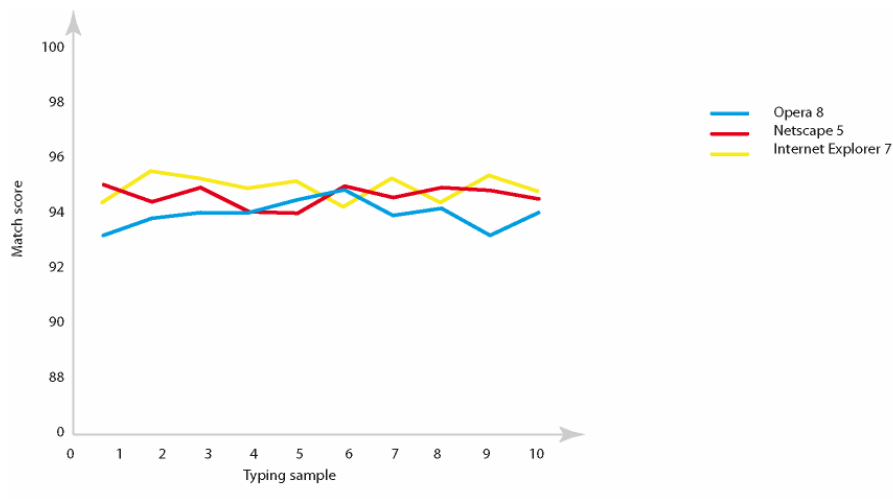


Fig.7-6 Match scores reached by different browsers while authenticating to a biometric profile created with Netscape

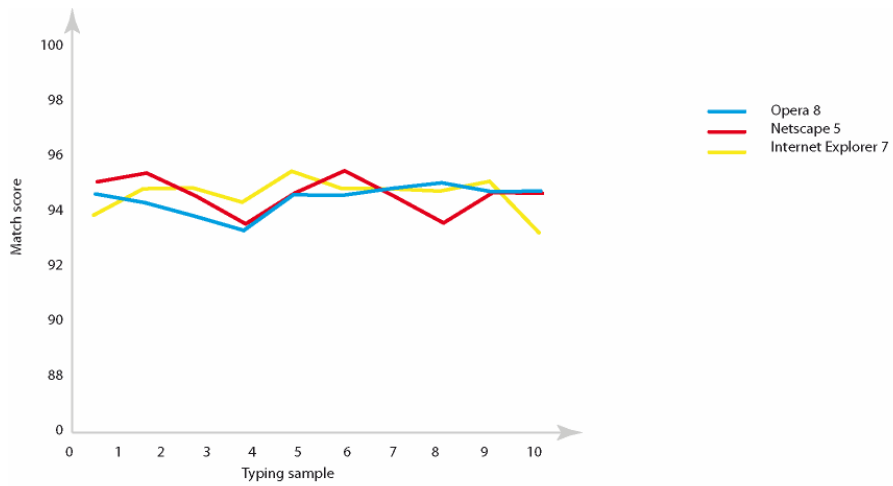


Fig.7-7 Matching scores reached by different browsers while authenticating to a biometric profile created with Internet Explorer

The average authentication match score of each browser against every other browser in the test is presented below.

Browser	Opera 8.0.54	Netscape 5.0	IE 7
Opera 8.0.54	95,15	94,46	94,12
Netscape 5.0	95,22	94,28	94,46
IE 7	97,12	95,24	94,64

Table 7-4 Results of the enrolment – authentication analysis

Test results: although the best result was achieved by enrolling with Opera and authentication with Internet Explorer, we must also consider the fact that the differences are not so big and that the biometric method has also heuristic components which may slightly influence the test result. We consider that all the browsers tested are fully compatible with typing cadence.

#### **7.4 Hardware problems (different keyboards)**

Biometric systems based on typing behaviour have a strong dependency on the hardware sensor used in the process of enrolment and authentication. In this case, the sensors are different types of keyboard, which differ either in the way in which they are built (different keyboard layouts, different degrees of ergonomics) or in the basic technology used to send key press/release signals to the computer. Additionally, users tend to have a preference towards one of the keyboard types and thus show a changed typing behaviour on an unusual keyboard.

A test conducted in 2006 at the University of Regensburg has shown that users that have enrolled on a certain type of keyboard and later tried to authenticate on a different keyboard had difficulties in gaining access to the system.

Therefore, it is necessary to make a test with several keyboard types. This test must answer to the following questions:

- How does the match score fluctuate when a user changes the keyboard after enrolment?
- How many typing samples have to be included in the biometric profile so that the profile accepts more keyboards?
- Is it necessary to use different keyboard profiles?
- Which keyboards are similar and can share the same profile?
- How should a mixed keyboard profile be determined?

#### 7.4.1 Test procedure

For the test procedure, we used four different keyboard types:

- Standard PS2 keyboard: The reason for using this type of keyboard in the test was the fact that assumedly the PS2 connector may cause quality problems due to its age. Additionally to that, this keyboard had an older system of keys, similar to a typing machine.
- Standard USB keyboard: This is a modern keyboard and was used to check the possible differences to PS2.
- Notebook keyboard: Despite of the fact that the keyboard layout of a notebook is the same as at a normal keyboard, the keys on a notebook are smaller. The keys can be more easily pressed, but they are not as robust as the keys from a pc keyboard.
- Wireless keyboard: The wireless keyboard used was very similar to the USB device, with the difference that it had a cordless connection to the computer. The assumption was that this keyboard would cause problems due to the delay between the built in wireless sender and the receiver connected to the computer.

The quality of the four biometric sensors was tested with the following procedure: the test persons were asked to input 50 sentences with each keyboard, out of which the first 30 were used to create a biometrical profile and other 20 were used for authentication.

The reason for choosing 30 samples for enrolment lies in the empirical experiments of (Achatz 2006) who used a similar test system to determine the EER dependence on the number of samples used for enrolment. His results showed that a good EER can be obtained starting from 30 user samples.

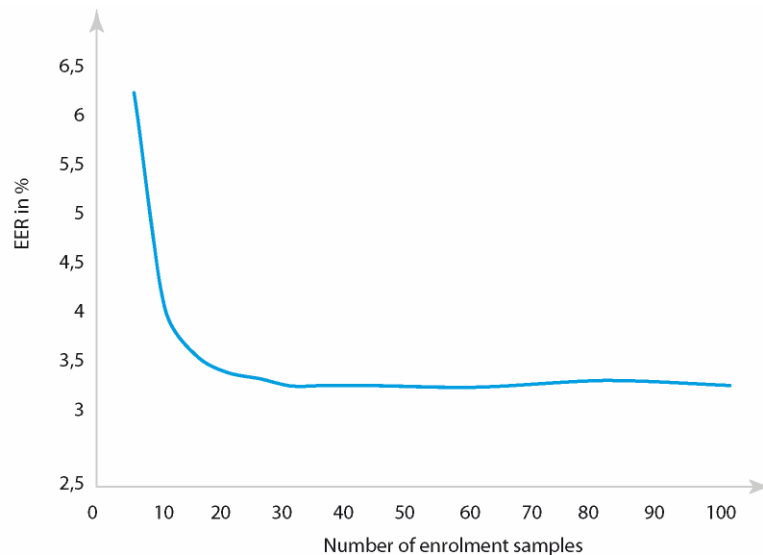


Fig. 7-8 EER dependence of the number of enrolment samples (Achatz 2006)

As the number of the biometric samples that one user has to type is very high (200 samples) and requires a lot of time and effort, it is important to make a pre-filtering of the users allowed to participate in this test.

The selection criteria are divided in two groups:

- Number of fingers used for typing: we have used a rough division between two finger typist and 10 finger typists;
- The user's experience with different types of keyboards: here we tried to include persons that have experience with all the tested keyboards, as well as persons that are used to only one type of keyboard.

#### 7.4.2 Expected results

The test was divided in two phases:

Phase 1: Enrolment on one keyboard and authentication on all other keyboards.

Phase 2: Mixed enrolment made on all keyboards and authentication on one particular keyboard.

In order to have a good comparison basis, a set of assumptions was made, which are to be confirmed or rejected by the test.

Firstly, it was assumed that a keyboard change would cause losses in the recognition capacity of the biometric method. A question was how many typing samples are necessary to decrease FRR to its previous level.

A second assumption regards the authentication both with and without template adaption. In the normal case, the biometric method adapts the template after every successful authentication attempt in order to avoid aging problems (see chapter 8). It was assumed that without the template adaption the recognition rate would decrease, thus leading to a high FRR.

The third assumption was that the keyboards are not compatible with each other. The expected result was that upon a change of keyboard, the biometric method would return a decreased match score for the next authentication procedure. The 50% line shows a hypothetic threshold used for these assumptions.

The results were plotted in the following graph:

- X - axis: the number of the samples used after the enrolment (1, 2, 3...)
- Y - axis: the match score was reached upon the authentication with that particular typing sample.

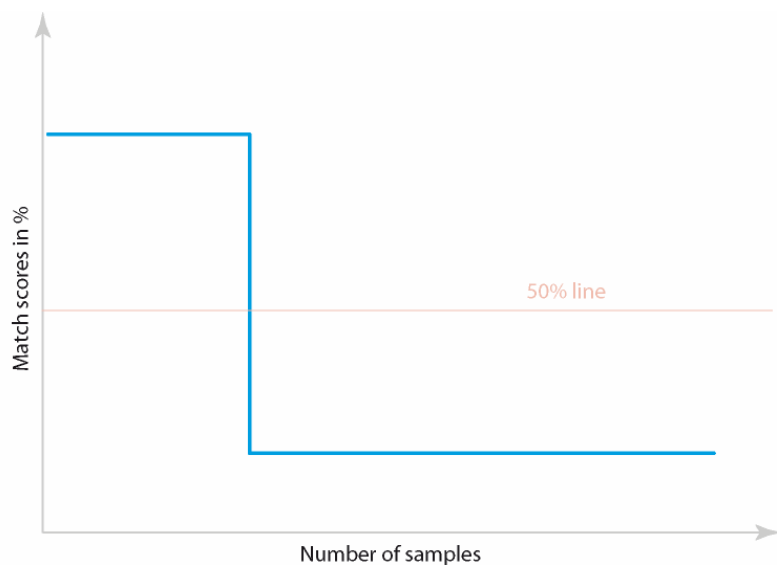


Fig. 7-9 Match scores by keyboard change without adaption

A return to good recognition values in the process of authentication is achieved only if the template is adapted with a number of  $N$  samples from the new keyboard.

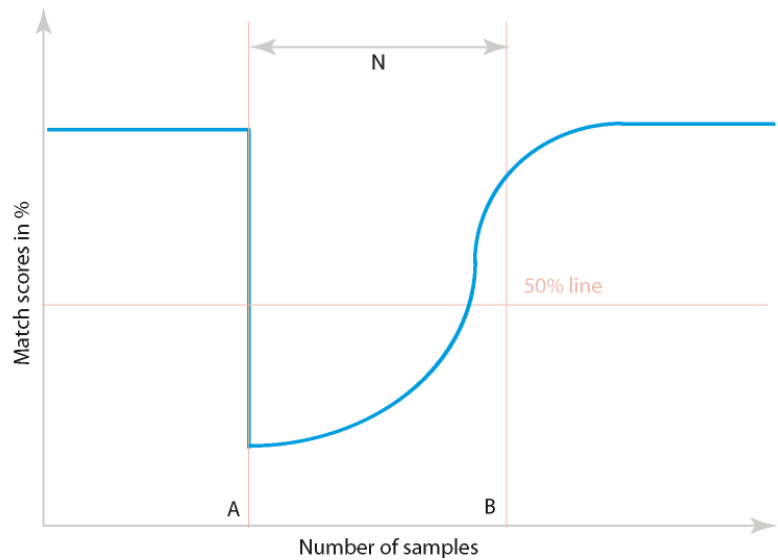


Fig. 7-10 Adaption of the template leads to higher match scores

In case of an enrolment process made on more keyboards, it was expected that the authentication with one of the keyboards from the test would lead to a smaller match score which would still be over the threshold. This is the expected graph for a change of keyboard without any template adaption:

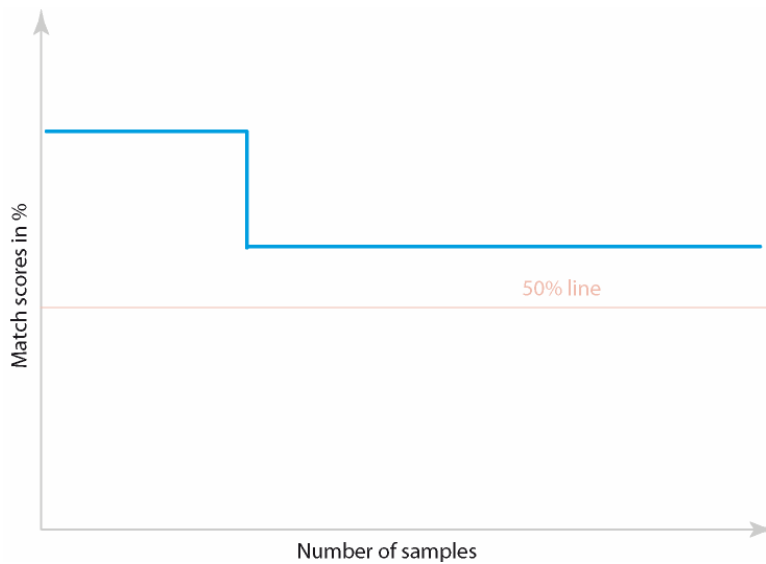


Fig. 7-11 Authentication to a multi-keyboard enrolment template without adaption

When the template adaption is enabled, we expect a change of the match score which would still lie over the threshold. After  $M$  samples (where  $M < N$ ), the recognition rate will become the same as before.

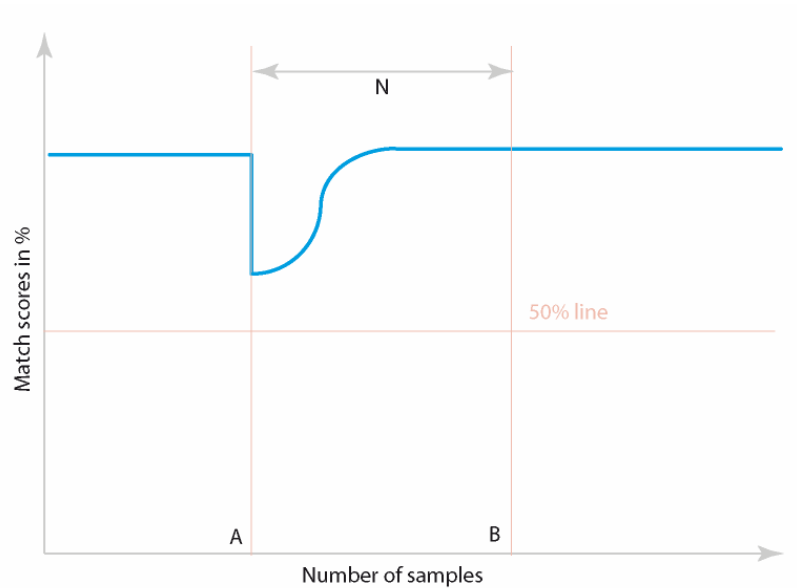


Fig. 7-12 Authentication to a multi-keyboard enrolment template without adaption

### 7.4.3 Test results

The following configurations were used in order to verify the aforementioned assumptions:

1. Test with the same keyboard but without any template adaption

For this test, the first 30 typing samples were used to create a user profile and the rest of 20 samples from the same keyboard for the authentication process. The results obtained by each tester were then averaged and arranged in the following graph:

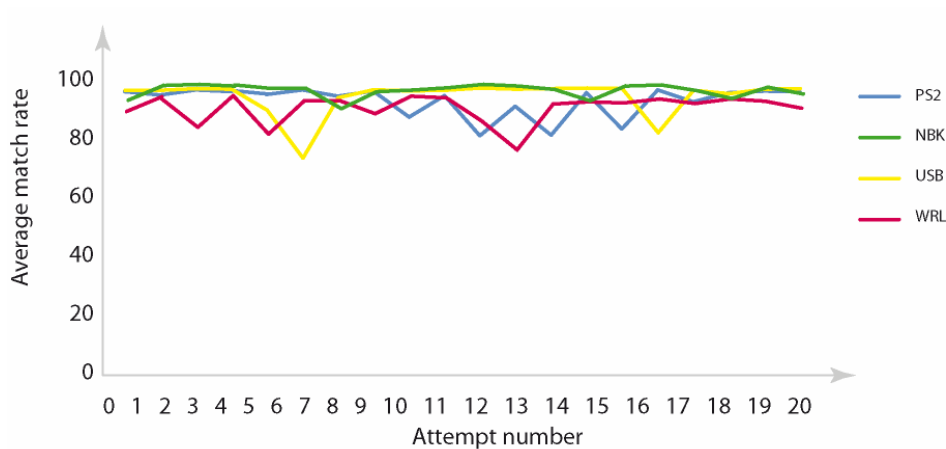


Fig. 7-13 Quality of the typing samples without the adaption

Test results: as expected, even without template adaption the samples deliver good results while authenticating to the profile created with the same keyboard. The numbers on the X axis show the number of the authentication attempt, while on Y axis displays the match score achieved.

## 2. Test with the different keyboards but without any template adaption

For the next test, the users enrolled on one keyboard. Then all the 50 samples from the other keyboards were used to authenticate against that type of keyboard.

For the case of the wireless keyboard (light blue in the graphic), this test followed the procedure:

- Enrolment with 30 samples from the wireless keyboard;
- Authentication with 50 samples from the USB, PS2 and notebook keyboard;
- Average of the results on these types of keyboard;
- Average of the results of all the test users.

This test shows how compatible one type of keyboard is with all other keyboard types from the test.

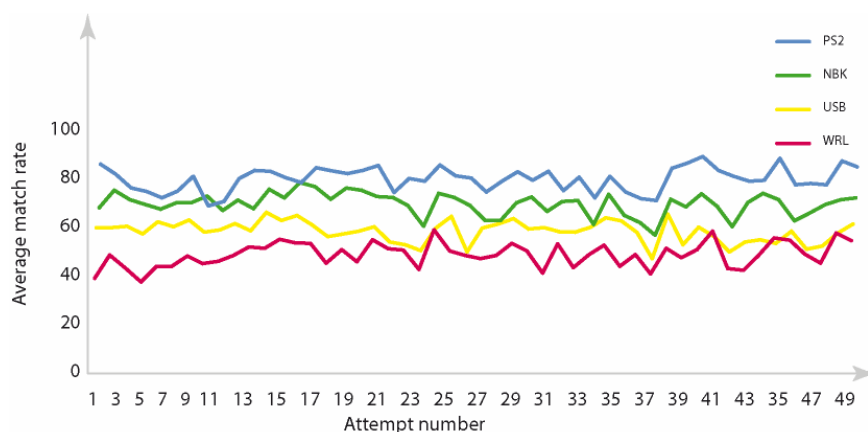


Fig. 7-14 Different keyboards without adaption

Test results: this test demonstrates that the match scores have decreased with ca. 20 % compared to the previous test, especially in the case of USB and wireless keyboards. Although the typing behaviour is the same (we used the same typing samples as before), the recognition rates decreased drastically.

The conclusion is that the keyboards are not compatible with each other; upon enrolling with a certain keyboard, the user must authenticate with the same type of keyboard. For each keyboard we need a different profile and template.

### 3. Test with the same keyboard and template adaption

The same test procedure is repeated, this time with template adaption switched on.

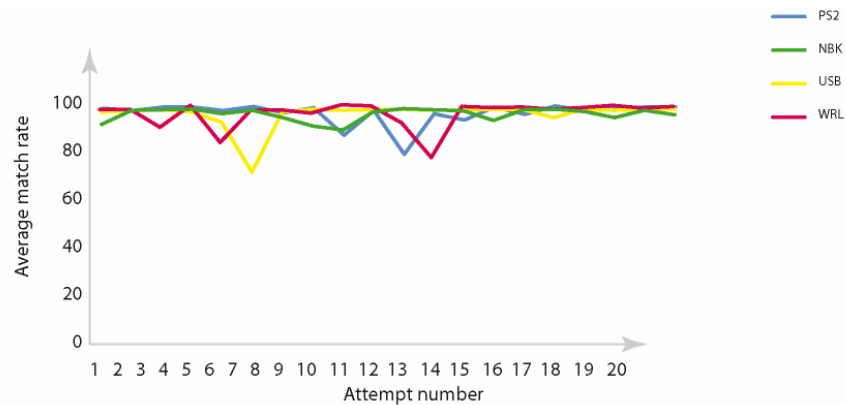


Fig. 7-15 Template adaption

Test results: in this case, the high quality of the match scores does not leave much place for improvement that is why the results are similar to the ones obtained in test 1.

### 4. Test with different keyboards and template adaption

In this case we try to determine the number of samples  $N$  the template needs to assimilate in order to accept the typing samples of the new keyboard.

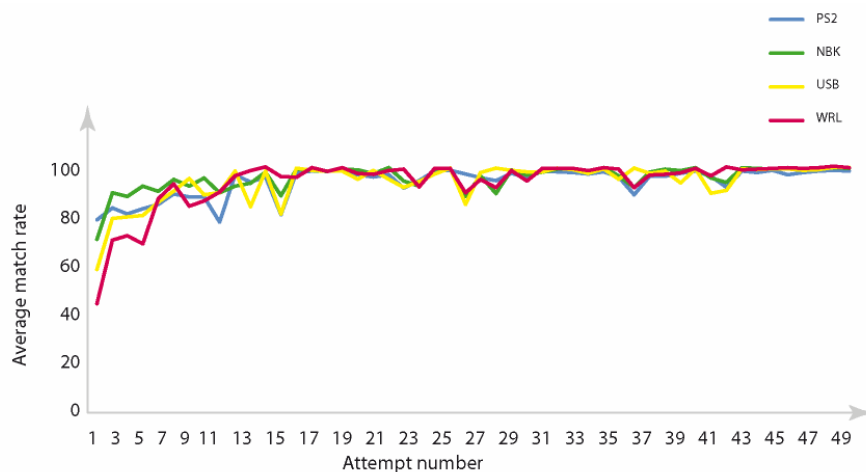


Fig. 7-16 Template adaption with multiple keyboards

In this case, the quality is increasing quickly. We notice that already after the third attempt, the match scores are high enough to allow correct user recognition. For a more secure statement, we can say that the user is correctly recognized starting from the 10<sup>th</sup> sample adapted.

Thus we determine the number of foreign samples that has to be added to template adaption in order to create a new profile:

$N = 1/3$  of the number of samples used to create the original profile

Then, the number of samples necessary to create a new profile for a new keyboard:

$$NoOfEnrolSamples = \frac{2}{3} SamplesFromOldKeyboard + \frac{1}{3} SamplesFromTheNewKeyboard(N)$$

For our example:

$$N = \frac{NoOfEnrolSamples}{3} = \frac{30}{3} = 10 TypingSamples$$

And:

$$NoOfEnrolSamples = 20 OldSamples + N(10 NewSamples)$$

5. Test with a profile made from all keyboards without any adaption:

In this new test, we try to enrol the user with a mixed profile created by samples from different keyboards and then try to authenticate all the remaining samples against it.

For the wireless keyboard (red), this process would have the following phases:

- Enrolment with 20 samples: 5 samples from each keyboard type;
- Authentication with 45 samples from each keyboard type;
- Average of the all 45 match scores from the 4 keyboard types;
- Average of all users from this test.

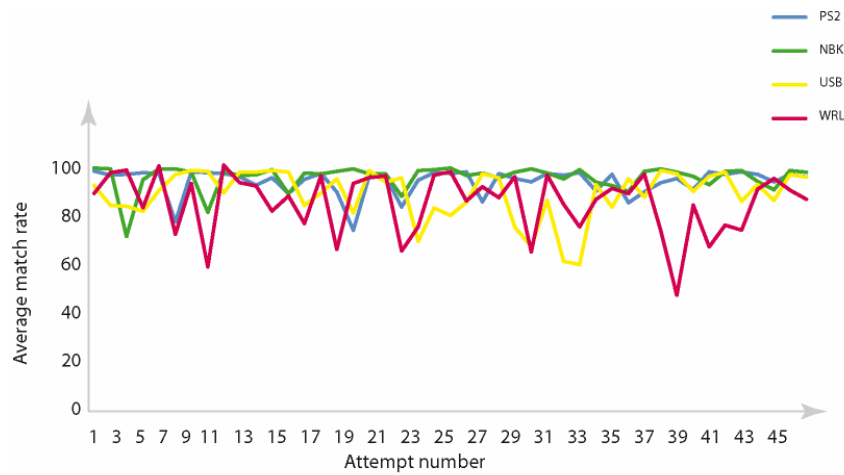


Fig. 7-17 Mixed profile while attempting to log in with all keyboards

As expected, the match scores are slightly lower; nevertheless the users would have managed to authenticate against this profile with all possible keyboards. The only keyboard which shows outliers is the wireless keyboard, most likely due to the fact that the wireless channel is slower than the cable.

#### 6. Test with a profile made from all keyboards and with adaption:

In this test, we determine the number of samples  $M$  necessary for the template to recognize the keyboards from the test. As previous results showed that even without adaption a profile created from samples belonging to all kinds of keyboards allows the user to authenticate with all keyboards from the profile, it is not necessary to repeat this test. From the previous test we deduct that  $M$  is equal to zero.

#### 7. Test of FAR and FRR for the mixed profile with adaption:

The fact that a mixed profile allows the user to log in with any keyboard from the profile raises the suspicion that the profile (template) would be broad enough to even allow other users to access it. To check this, we calculate the FAR and FRR curves for the mixed profile in the following way:

- create a mixed profile for each user (5 samples from each keyboard type);
- match all the rest of his samples with his profile;
- match all the samples of other users with his profile;
- create the FRR curve from the match scores of his samples;

- create the FAR curve from the match scores of the other users;
- average the values over all users.

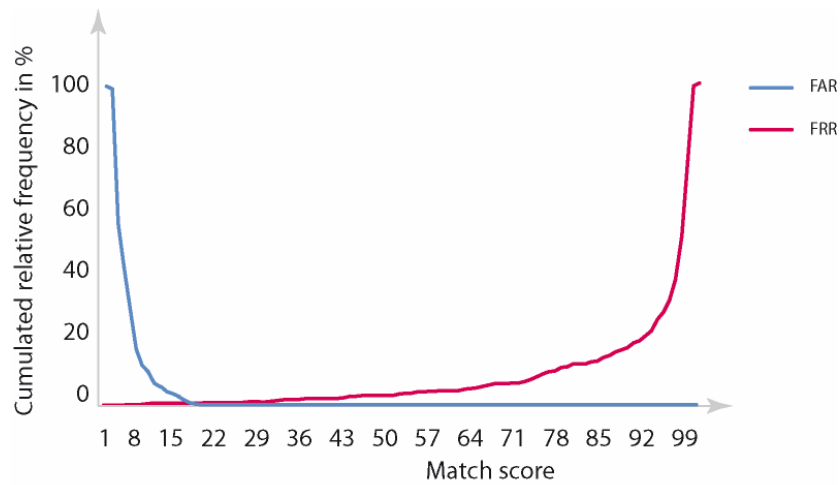


Fig. 7-18 FAR and FRR curves of the mixed profile

The graph shows that the supposition that the profile would not be strong enough in the case of more keyboards has proved wrong, as the EER is less than 1%.

#### 7.4.4 Conclusion

The software tests have shown that, while designing a client recorder, we must take into consideration the browser on which it is running, the technology used to create the recorder and the operating system with its particularities.

In the same way, the hardware tests have proved that, for each kind of keyboard used, a new biometrical template has to be created. Nevertheless, the enrolment for this template can be shorter with 66 %, as less typing samples are needed for the new profile.

## **8 AGING OF BIOMETRIC FEATURES**

---

As all persons age with the time, so do their biometric features. This aging process can be observed over longer periods of time (for image-based biometrics) or it can happen very fast (in case of behaviour biometrics). The biometric method used for this system belongs to the last category, which makes it important to research which of its features are influenced by the aging process.

---

### **8.1 Aging of the reference template**

Template aging is a phenomenon in biometrics that includes a continuous deterioration of the system's identification quality. The reason for this lies in the gradual change of the biometric features. The feature is less and less congruent with the one used at the point of enrolment, which results in a worsening of the match rate until the point where it rarely trespasses the threshold and thus leads to false rejections of the user.

Changes of the feature occur to a greater or lesser extent in all biometric methods. Even with fingerprints, Pontus Hymér found out that they are subject to an aging process. (Hymér 2005) Although the pattern is very resistant, small changes that are based on humidity, dirt, abrasion or minor injuries constantly alter it. Particularly craftsmen or other persons that put a strain on their hands in their daily work are affected. Sometimes, the abrasion is so severe that it is not possible to create a proper sample anymore. Different skin diseases also lead to a fast change in the shape of the fingerprint.

Another biometrics strongly affected by template aging is face recognition. The looks of a human face change rather often depending on different influences. These can be temporary changes like make up or beard growth, as well as the increasing number of wrinkles due to old age. Identification quality is also influenced by weight loss or weight increase. Stable changes are for example scars etc.

The human voice as well strongly varies in the course of a person's life. Hoarseness or a common cold can cause large differences to the reference pattern. In the time of puberty, boys are subject to a complete change of the pitch of the voice.

Such developments can also alter a person's retina. Illnesses like the glaucoma or diabetes can cause such damage to the retina that identification and authentication become difficult. (Betschart 2005)

Many paths have already been gone in order to curb template aging. Several authors suggest the creation of a new reference pattern, a so-called re-enrolment process. The question with this method is when and how often this procedure should be repeated. In any case, it would cause additional effort for the user and the administrator (in case of a supervised enrolment).

Another approach is a possible adaptation of the template. Nolde suggests to "[...] automatically adapt the corresponding reference patterns to the changed biometric feature after a successful identification" (Nolde 2002)

This process continuously modifies the template and adapts it to small changes, without causing the user to create a new profile. This method is used particularly in typing cadence. Many authors have noticed at an early stage that if the same sequence, a so-called predefined text, is used for authentication, this sequence runs in with the time. Random sampling by Bakdi has shown an early increase in speed (number of keystrokes per minute) with test persons. (Bakdi 2007)

To what extent template aging causes problems to typing cadence biometrics has hardly been researched until now. It is known that typing runs in with the time, but there have been no analyses as to which characteristics in particular are changing the most. As different typing cadence methods use different characteristics, in this work we will concentrate on the Psylock method (Bartmann 2004) and its characteristics.

## **8.2 Experimental setup**

The goal of the experiment was to identify and research the changes that occur to characteristics of typing cadence.

For this purpose, several trial participants had to deliver typing samples over a longer period of time. There were 18 volunteers that gave 2-3 samples every day for the period of 16 days. The data capturing was carried out by a special web application that can record typing cadence and store it persistently in a database. A tool especially developed for the experiment calculated the biometric characteristics from the stored raw data and determined the respective daily average per user and an

overall daily average for all users. The same tool also visualised the data showing the development of the characteristics over the 16 days.

The following figure shows the experimental setup and feature extraction.

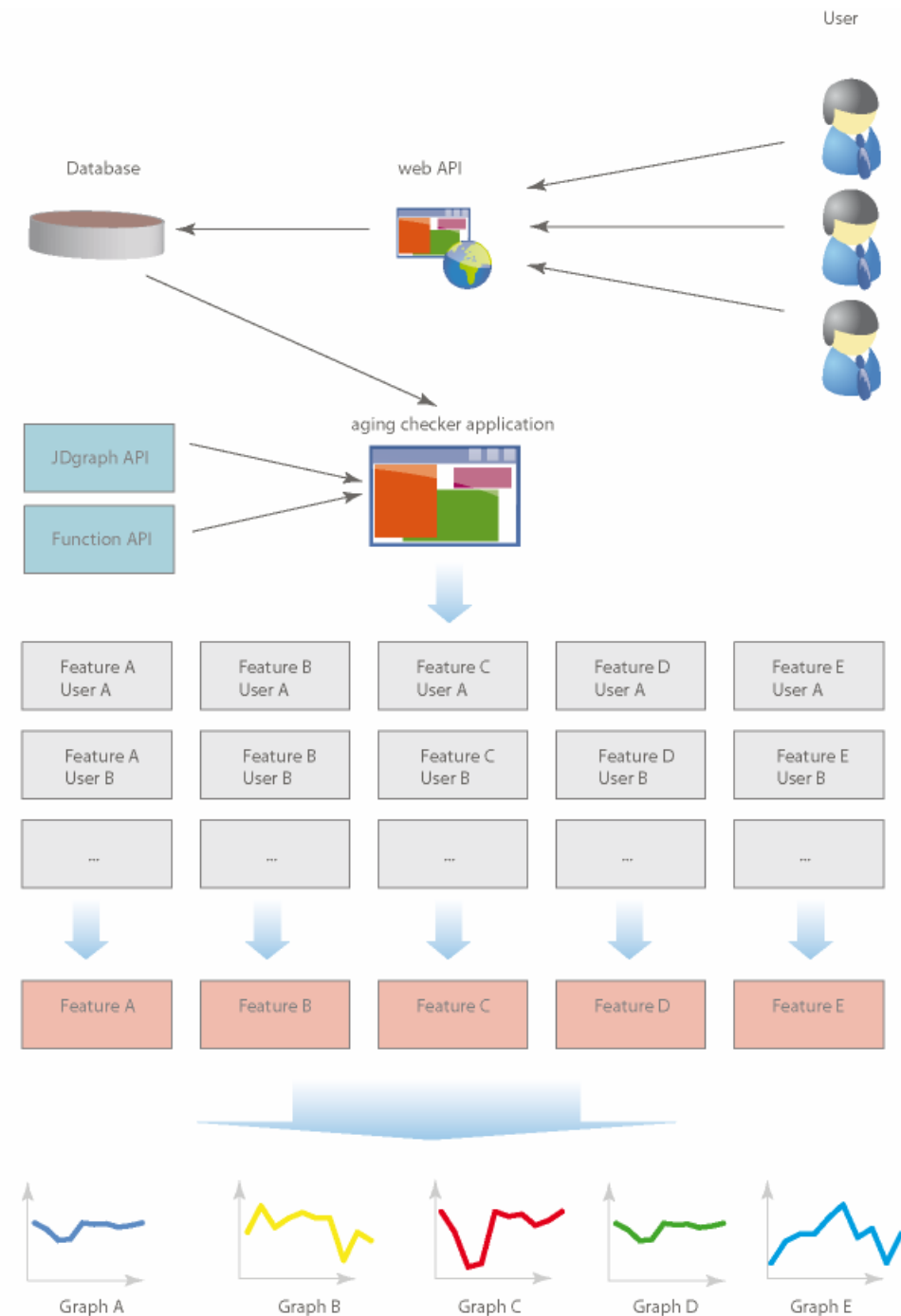


Fig. 8-1 Experimental setup to determine the aging process of typing behaviour biometric

As mentioned, every day the participants delivered typing samples over an input mask in an installed web API. The experiment uses a predefined text, so users always had to type the same German sentence as follows:

„Ich bin der Meinung, die richtige Antwort lautet:”

(“After my consideration, the correct answer is:”)

The web API records the user’s typing cadence and saves it as raw data in the database. The raw data has the following format:

```
"2006_12_27_09:58:27&016v9999&073v0160&016^0060&073^0000&067v0170&067^0060&072v0151&072^0120&032v0140&032^0080&066v0180 . . . "
```

The first part gives information on the exact time of the sample’s creation. Follows the key code that tells which key an event occurred at. The next symbol specifies the kind of event, the “v” stands for a press event and the „^” for a release event. The last number before the next “&” gives the duration of the respective event.

The aging checker application takes this raw data about every user and every day from the database. Before any statements can be made about characteristics, however, the data has to be brought in another form. A function of the function API extracts information from the raw data, divides it in key codes, single times and events and hands it over to three different arrays. The result is the following:

- An array containing the key codes of all keys in the order of event appearance would look like this:

```
k = Array(016, 073, 016, 073, 067, 067, 072, 072, 032, 032, 066)
```

- An array where the kinds of events are listed chronologically:

```
i = Array(v, v, ^, ^, v, ^, v, ^, v, ^, v)
```

- An array with the time in milliseconds that corresponds to the events:

```
z = Array(9999, 0160, 0060, 0000, 0170, 0060, 0151, 0120, 0140, 0080, 0180)
```

Lastly, the features are calculated from the so arranged data. This is done by the function API. The next step is to average over every characteristic per day and user saved in CSV files by the API. This is done by the aging checker application. Every user and characteristic is stored in its own CSV file that contains the development of the feature in question over the entire period of time.

Next, the aging checker application processes the CSV files one after the other for every characteristic and calculates the daily averages of all users per characteristic. The result is the overall development for every single characteristic, which is then saved in a separate Excel file.

The open source software library of JdGraph visualises the results.

### **8.3 Feature extraction**

One of the biggest challenges of a method for pattern recognition is the extraction of adequate features that can clearly identify users and delineate them from others. Many methods are limited to the pressure and transition times.

- Pressure time:

The pressure time is the time that passed between the pressing of a key and until its release.

- Transition time:

The transition time is the time that passes between the releases of a key until the pressing of the next key.

Theoretically, these two times could be the basis for all characteristic features of typing dynamics, but this would presume an enormous amount of learning data. The result would be an extensive and elaborate enrolment, which would lower user acceptance.

In order to achieve a sufficient identification quality despite a small amount of learning data, it is necessary to derive additional selective features from the raw data that would balance the small amount of samples. The raw data is filtered for implicit information and this information is pre-processed in order to be optimized for further handling. The feature families gained from calculation each represent a different aspect of typing cadence and thus contribute to an amelioration of the system's quality.

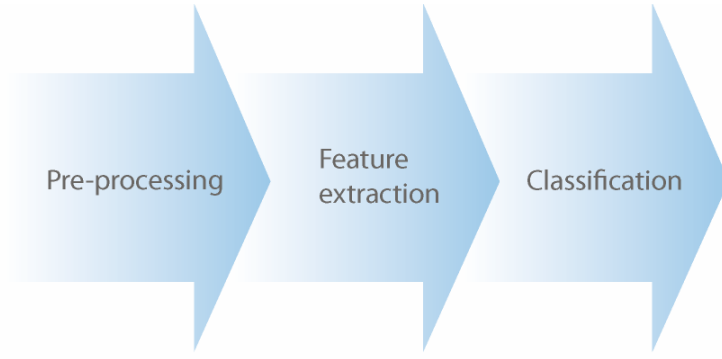


Fig. 8-2 The feature processing chain (Bakdi 2007)

For Psylock, the focus of feature collection is laid on two groups of features. The first derives directly from the pressure and transition times and makes the group of the time dependent features. The second contains time independent feature families and is based not on time but exclusively on events. (Bakdi 2007)

## 8.4 Time dependent features

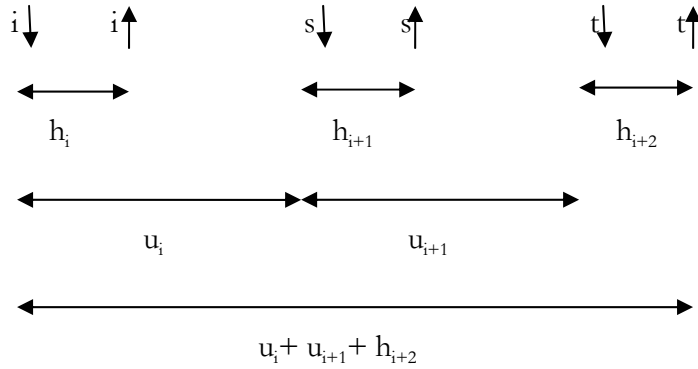
### 8.4.1 N-segment duration

#### 8.4.1.1 Calculation

The n-segment duration is the time that is needed for the creation of a certain character string, also called n-Gramm. It is important to know that, for the input of an n-Gramm, it is necessary to take into consideration every event on the keyboard, i.e. also the pressing of a Shift key in order to make a capital letter or a colon. This way, it is possible that for n-Gramms of the same length, a different number of key events is necessary, depending on the amount of capital letters or special characters. Therefore, this work considers the entire sequence of key events, also called a segment of the length  $n$  ( $n$  being the number of key events), or n-segment, independent on the resulting character string.  $S[n]$  is the code for n-segment duration. The following formula is used by Psylock for the calculation of  $S[n]$ .  $h$  is the vector of the pressure durations and  $u$  the vector of the transition durations.

$$\chi_{(S[n]),i} = h_{i+n-1} + \sum_{l=i}^{i+n-2} u_l$$

If a user wants to type a word, he has to press and release keys. The pressing of a key is symbolised by an arrow down and the release by an arrow up. When the word *ist* is typed, the event chain would ideally look in the following way:



The  $n$ -segment duration in this case is the sum of the transition times  $u_i$  and  $u_{i+1}$  and the last pressure time  $h_{i+2}$ . (Bakdi 2007)

Another method to calculate the overall time is based on the fact that, as mentioned above, the tool extracts from the raw data the lengths of the separate pressure events. By adding the separate times of the array  $z$ , the result is the overall time that was needed to type in the character string. The formula is this:

$$\sum_i^n z[i]$$

where  $z[i]$  stands for the separate times (positions in the array).

#### 8.4.1.2 Expectations

This experiment uses a predefined text; the trial participants have to type the same sentence again and again. Therefore it is to be expected that the persons will quickly get used to the process and will know the exact character string after a few days at the latest. Fluent typing should be possible from that time on.

Furthermore, the users repeat the same movements of hands and fingers when typing, so it is possible that a kind of conditioning takes place and the users type the sentence in question almost automatically. This mechanization should strongly contribute to a faster typing time.

For these reasons, it is to be expected for the n-segment duration to decrease enormously and then to remain at a constant level. The last will occur because at some point, the user has reached his personal speed limit and can not increase typing speed any longer.

It is also to be expected that fluctuations of the user's day's form will occur just as they are found with other active biometric methods.

The development should look the following way:

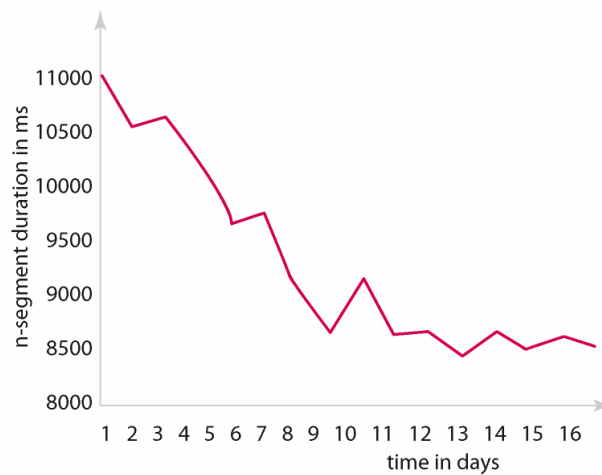


Fig. 8-3 Expected development of the n-segment duration

#### 8.4.1.3 Analysis

This graph shows the actual development of the n-segment duration over the entire time of the experiment.

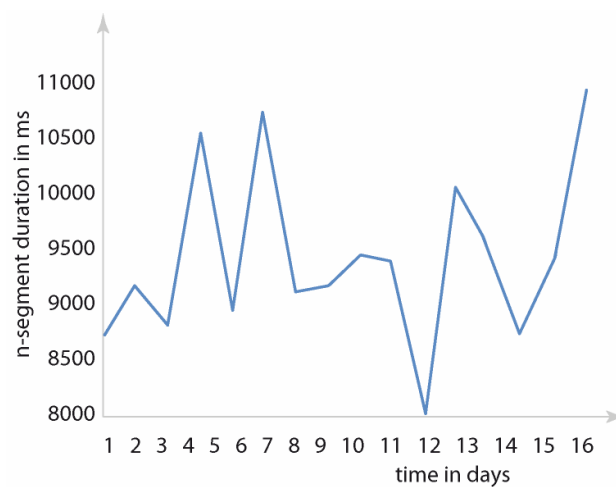


Fig. 8-4 Actual development of n-segment duration

The values on the x-axis stand for the day of the experiment and the y-axis gives the n-segment duration in milliseconds. The values of the duration are given by the average times of all users over the entire day.

The curve shows fluctuations of the n-segment duration, which differs in two seconds maximum from day to day. This fact is not surprising as such irregularities are normal with an active biometric feature.

More interesting is the observation that the n-segment duration slightly increases with the time. The users seem to become slower in typing, which is contrary to the aforementioned expectations. The fact of getting used to the typing sentence does not have any influence upon the overall time that is needed for typing. The reasons for this might be the fact that users initially try to type the sentence as fast as possible. After getting used to the process, however, they need less concentration and they start typing in a comfortable speed that is actually normal for them.

## 8.4.2 Speed

### 8.4.2.1 Calculation

The calculation of typing speed  $G[n]$  is closely related to the n-segment duration and is directly derived from it. The following formula was used by Bakdi:

$$\chi_{(G[n])} = \left( \frac{n}{\chi_{(S[n]),1}}, \frac{n}{\chi_{(S[n]),2}}, \dots, \frac{n}{\chi_{(S[n]),r-n+1}} \right)$$

Analogue to the n-segment duration,  $n$  stands for the overall number of key events. Following the physics formula for speed, which says that speed is distance divided by time, we divide the “distance”  $n$  through the “time” n-segment duration of the individual segments. The result is the speed with which a user typed the respective segments of the input text. (Bakdi 2007)

The process of raw data processing used in this work slightly differs from Bakdi, which leads to a modified formula:

$$\chi_{(G[n])} = \frac{\sum_i^n \frac{n}{z[i]}}{n}$$

The overall number of key events is divided through the single events times ( $\bar{x}[i]$ ) and the results are added. In order to achieve the average of speed, the result is divided through the number of events  $n$ .

#### 8.4.2.2 Expectation

If the n-segment duration continuously decreases until the point where it settles at a constant value, it is to be expected that the separate times will decrease as well. This would result in an augmentation of speed which would continue until this too settles at a stable level.

The following curve progression is to be expected:

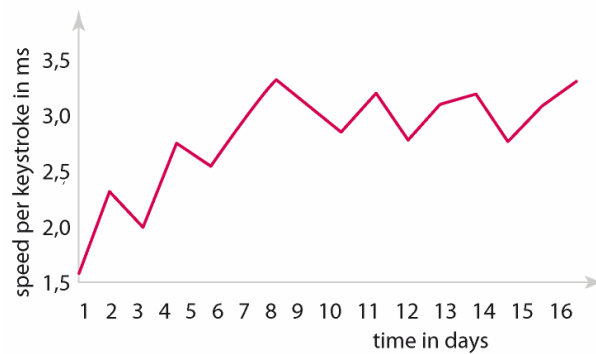


Fig. 8-5 Expected development of speed

#### 8.4.2.3 Analysis

The graph below shows the actual development of the feature family speed. The x-axis shows the time in days, the y-axis the average speed per keystroke of all users in milliseconds.

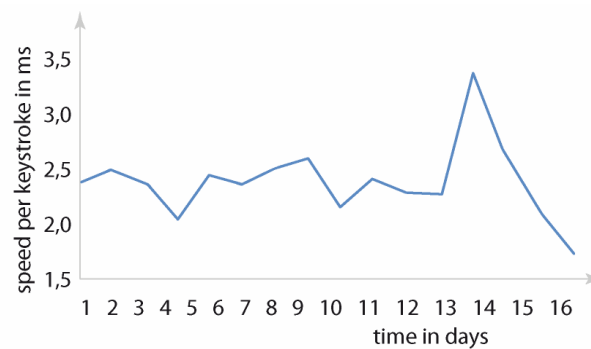


Fig. 8-6 Actual development of speed

The graph shows that except days 13 and 16 the speed remains relatively constant and can be therefore counted as a very stable typing characteristic. It is also subject to slight fluctuations, but these do not surpass 0.4 units.

The expected increase in speed did not occur and one can even observe a slight decrease of speed. The reason for this is that the speed is reciprocally proportional to the n-segment duration and therefore an increase in typing duration automatically goes along with a decrease of speed.

This result can be also influenced by the fact that a big part of the users considered for the tests were already familiar with the sentence which they used before, therefore the phase of getting acquainted with the method cannot be observed.

### 8.4.3 Outliers

#### 8.4.3.1 Calculations

Exceptions (here also called outliers) are particularly long and eye-catching pauses in a user's typing cadence.

Almost all previous research on typing cadence did not especially focus on exceptions. In most of the cases, they were ignored altogether. Psylock, on the contrary, models them as an own feature family. In the following formulas,  $h$  and  $u$  stand for the pressure and transition durations before removal of the outliers. This is a particular case as outliers are usually removed in a pre-processing phase, when biometric samples are checked for qualitative features and possible disturbances (e.g. missing key events, 0-value times or, in our case, outliers) are removed in order to guarantee a proper biometric measurement.

In our case,  $b_{\mu(H)}$  and  $b_{\mu(U)}$  respectively  $b_{\sigma(H)}$  and  $b_{\sigma(U)}$  are the vectors of the average values or rather the standard deviations which have been calculated in a previous step and stored as a part of the typing profile. Outliers of the pressure durations are called  $\chi_{(A(H))}$  and the outliers of the transitions durations  $\chi_{(A(U))}$ .

The following formulas model the two feature families:

$$\chi_{(A(H))} = \left( \frac{h_1 - b_{\mu(H),1}}{b_{\sigma(H),1}}, \dots, \frac{h_r - b_{\mu(H),r}}{b_{\sigma(H),r}} \right)$$

and

$$\chi_{(A(U))} = \left( \frac{h_1 - b_{\mu(U),1}}{b_{\sigma(U),1}}, \dots, \frac{h_r - b_{\mu(U),r}}{b_{\sigma(U),r}} \right)$$

This procedure is derived from Grubb's test statistics (NIST eHandbook 2008) and measures the difference between the observation  $h_i$  or  $m_i$  and the average value  $\mu_{h,i}$  or  $\mu_{m,i}$  in units of standard deviation  $\sigma_{h,i}$  or  $\sigma_{m,i}$ . The results give a measure for the probability that a respective time interval is an extreme.

From this, one can determine places in the typing model where outliers often occur at a certain user. The feature families have been proved very selective by (Bakdi 2007).

This work does not focus on the places in the typing model that tend to cause outliers, but the average probability of the occurrence of an extreme; therefore we do not distinguish between pressure and transition durations. The basis for the calculation is the following formula:

$$\chi_{(A)} = \left( \frac{\sum_i^n \frac{z[i] - \mu_z}{\sigma_z}}{n} \right)$$

The probabilities of outliers are added and averaged.  $\mu_z$  or  $\sigma_z$  stand for the average value or rather the standard deviation of the entire interval of the single times  $z$ .

#### 8.4.3.2 Expectations

In theory, it is possible to distinguish two kinds of outliers. There are outliers that characterize the user typical typing cadence; they always occur at the same place and are closely related to the user's basic typing cadence. The second kind are situational or knowledge contingent outliers. They occur because the typist is unclear as to which word he has to type next or because he is distracted by outward influences. Illness or mental stress can also lead to an increased occurrence of longer pauses. Such outliers are irregular and cannot be anticipated; therefore their probable development can not be predicted.

The user specific outliers will probably remain constant if the person does not basically change her typing cadence. The only outliers that could decrease in frequency are knowledge based, as the user learns the typing model and does not have to look for the next word to type anymore.

It is probable that a slight decrease in the occurrence of outliers would result in time.

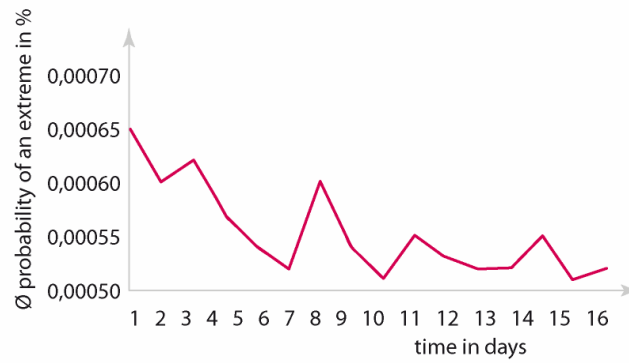


Fig. 8-7 Expected development of outliers

### 8.4.3.3 Analysis

The graph shows the actual development of outliers during the time of experiments. As before, the x-axis shows the time in days. The y-axis shows the average probability of an extreme. The average is taken from all users per day.

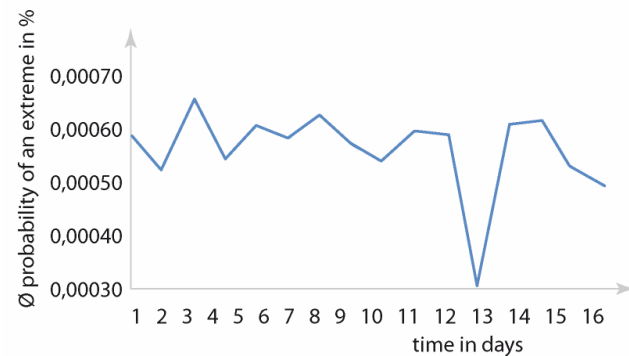


Fig. 8-8 Actual development of outliers

We see the fluctuation typical for active biometric features. Except day 12, it is slight and not noteworthy. The small amplitudes prove true the expectation that outliers mostly occur at the same place and remain stable.

Over the curve progression, there is a slight increase until day 7, after this the tendency goes down again. This can be explained by the fact that the participants get used to the typing model and that this fact causes a decrease in knowledge based outliers. This would definitely coincide with the expectations.

#### 8.4.4 Crossovers

##### 8.4.4.1 Calculation

(Bartmann 2001) introduced the feature family of crossovers and took it over for Psylock. The concept is based on the fact that there are several possibilities to type a character string. One way to type the word “cat” is to press and release “c”, then press and release “a”, then press and release “t”. The same string of characters results from releasing the “c” only after pressing “a”, which would cause a crossover.

A single crossover takes place when the “c” is released while the “a” is pressed. If the “c” is released only after the “a” is released, a double crossover takes place.

Bartmann found out the fact that when and how crossovers take place is very user specific and can be regarded as a feature of typing.

For the number of possible event sequences  $f(r)$  for a character sequence with the length  $r$ , while a crossover can occur at any place and to key is repeated, the formula is the following:

$$f(r) = \prod_{i=1}^r (2i-1) = \frac{(2r-1)!}{2^{(r-1)} * (r-1)!} > r!$$

Example: for  $r=10$ , the key space includes 654.729.075 possible events. In practice, also proved by Bakdi in a frequency analysis, there are only occurrences of single and double crossovers. Other crossovers with an order of  $\geq 3$  are negligible.

For this reason, there is another calculation for the number of possible events:

$$f'_{(r)} = \frac{(1+\sqrt{2})^r + (1-\sqrt{2})^r}{2}$$

For  $r = 10$ , the key space now includes 3363 different possibilities. The numbers are much lower than before, but still show an exponential growth.

It is necessary to mention that in most of the cases, there are no crossovers. If they occur, it is mostly the single crossover variant. The key space is enormously diminished by this fact.

The crossover feature family  $\ddot{ii}$  is modelled by the vector  $\chi_{(\ddot{ii})}$ :

$$\chi_{(\ddot{ii})} = (\chi_{(\ddot{ii}),1}, \chi_{(\ddot{ii}),2}, \dots, \chi_{(\ddot{ii}),r-1})$$

When there is no crossover,  $\chi_{(\ddot{ii}),i} = 0$ . Otherwise,  $\chi_{(\ddot{ii}),i}$  assumes a value that corresponds to the kind of crossover.

When typing the word “cat”, the result with no crossover would be (0,0), with one crossover (1,0) and with two crossovers (2,0).

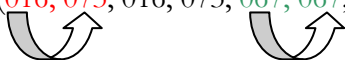
As all features, this family is strongly dependent on the user’s day’s form, which can lead to strong fluctuations of values. Nevertheless, they definitely contribute to a better recognition quality. (Bakdi 2007)

In this work, the research did not concentrate on the places where crossovers occurred but on the overall amount of crossovers in a given typing model. The procedure was the following:

The array  $k$  extracted from the raw data includes a list of all key codes (key events) that were effected one after the other. When a key code is listed in the array twice consecutively, it can be derived that there has been no other event between the pressing and the release of a key, and that therefore there has been no crossover. If a key code is followed by another, this means a crossover.

Example:

$k = \text{Array}(\textcolor{red}{016}, \textcolor{red}{073}, 016, 073, \textcolor{green}{067}, \textcolor{green}{067}, 072, 072, 032, 032, 066)$



The red marking shows a crossover, while there is no crossover at the green marking.

Every time when a key code is followed by a different one, the implemented algorithm increases the number of crossovers by one. This would mean that the transition from „067” to „072” is regarded as a crossover as well, but the overall number of crossovers is noticeably larger when there are real crossovers taking place. This makes it possible to observe and analyze an overall development.

#### 8.4.4.2 Expectations

As before, the values slightly fluctuate with crossovers as well, but they should remain relatively stable and settle down at a certain level. It is possible that an increase in typing speed causes a higher number of crossovers, as the fast typing increase the possibility that pressure events take

place before the release of the previous key. As soon as the typing speed remains constant, crossovers will as well. The following development is probable:

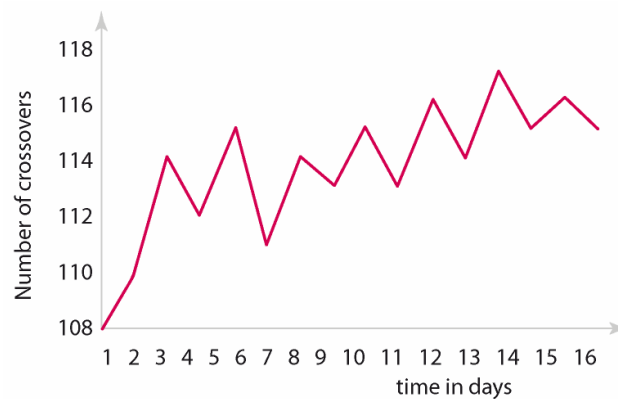


Fig. 8-9 Expected development of crossovers

#### 8.4.4.3 Analysis

The graph shows the actual development of crossovers over a time of 16 days. In analogy to the other graphs, the x-axis stands for the time in days. The y-axis shows the average number of crossovers of all persons. As mentioned before, the transition from one letter to the next is also assessed as crossover. Therefore the values of the y-axis do not show the exact number of crossovers, but of course the correct tendency. An analysis is fully possible with these values.

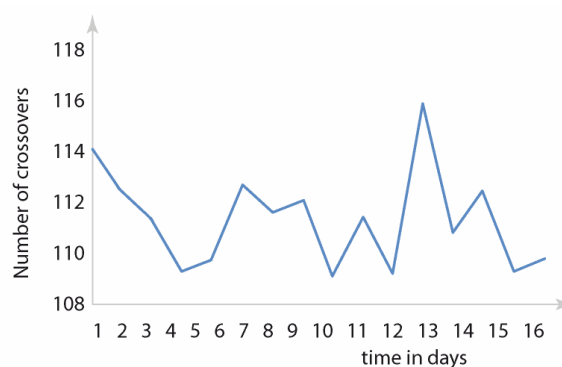


Fig. 8-10 Actual development of crossovers

The expected fluctuation proved by Bakdi can be also seen in this experiment. It is very small and mostly differs just by 2-3 crossovers. The maximum value between days 11 and 12 lies with 6 crossovers.

An exact analysis shows that the number of crossovers continuously decreases until day 11. Only then there is a short increase which immediately reverses. This confirms the downward drift.

This development is contrary to the expectations. These were that with a higher speed, users would make more crossovers. This development can be explained by the results of the n-segment duration and the typing speed. With the higher typing duration and the lower typing speed, the probability for hasty typing and crossovers decreases.

## 8.5 Time independent features

### 8.5.1 Typing mistakes and correction behaviour

#### 8.5.1.1 Calculation

In analogy to the feature family of outliers, typing mistakes are filtered with most typing cadence methods. Bakdi, however, found out that in combination with pressure and transition durations, typing mistakes and correction behaviour can be considered a characteristic feature of typing.

An experiment with ca. 200 users has shown that key events that had to do with mistakes of any kind accounted only for 4.3% of all key events. This low number makes it logical to unite all mistakes in one feature family  $F$  instead of dividing them after the kind of mistake. The entropy would be too small for the separate vectors of omissions, substitutions, commutations, insertion or corrections. (Bakdi 2007)

As these vectors do not give individually any significant information, we divide the sample  $S$  in more  $k[i]$  events. We distinguish between an ideal sample  $S^*$  (with the keys  $k^*[i]$ ) and a real sample  $S_x$  (using  $k_{x[i]}$ ).

The formula for calculating the differences between these two data is:

$$\mathcal{X}_{(S^*, S_x)} = \text{Levenstein}(\sum_i^n k_{[i]}^*, \sum_i^n k_{x[i]})$$

Let us start from the following example: if the users have to type the word “cat”, then the two parameters are:

$$S^* = c a t$$

and

$$S_x = c t a$$

In this case, the number of mistakes is 1, as the user has mistyped the letters “a” and “t”.

It is necessary to measure the Levenstein distance from this ideal to the actual sequence typed by the user. In order to generate the comparison value from the sample, single values are taken from the  $k$  and  $e$  arrays and arranged in a row.

The Levenstein algorithm always calculates the minimum number of editing operations necessary to change the user’s string of characters in a way that would make it correspond to the reference pattern again. All kinds of mistakes and corrections are taken into consideration and the result is a value with which the frequency of mistakes can be analyzed. (Sulzberger 2008)

### 8.5.1.2 Expectations

At the beginning, the user does not know the typing model too well, which makes typing mistakes more probable. After a short phase of settling in, the user should know the sequence well and make almost no typing mistakes caused by errors. However, if the same typing sequence is used over months, the decreasing concentration on the user’s side can again increase the frequency of mistakes based on carelessness and therefore the number of corrections. Slight fluctuations are to be expected that are dependent on the user’s day’s form.

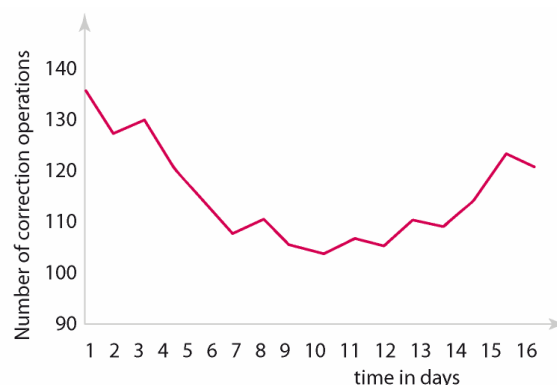


Fig. 8-11 Expected development of typing mistakes

### 8.5.1.3 Analysis

The graph shows the development of the average mistake behaviour of the test persons. The x-axis stands for the time in days, the y-axis for the number of operations that would be necessary in order to bring the sample into line with the ideal reference pattern.

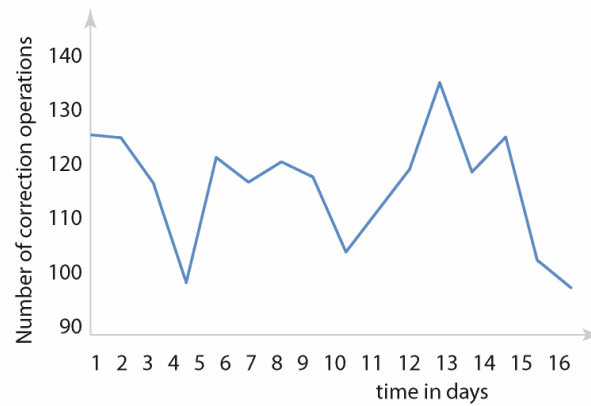


Fig. 8-12 Actual development of typing mistakes

The time independent features are also subject to the typical fluctuations in the measurement values. However, the difference between days is very high here. The values differ in up to 23 operations. The overall amplitude is significantly higher; it lies with 35 editing operations. This can be a clue to the fact that the mistake and correction behaviour is the feature with the strongest dependency on the user's day's form.

Initially, editing operations strongly decrease until day 9, in conformity to the expectations. Time and again there are high peaks that can be attributed to absent-mindedness of the users. Expect some outliers on days 12 and 14, the forecasted results occur and the overall number of mistakes decreases.

## 8.6 Conclusions

The experiments confirm the unsteadiness of features in an active biometric feature. All features under analysis have shown daily fluctuations. At some, such as outliers, crossovers and speed, these fluctuations are very slight, with high unexpected peaks on certain days. The big challenge for an identification system based on a behavioural biometric is to absorb these irregularities and still deliver excellent identification results. Otherwise, the method will not resist the practical test.

In order to research in how far typing cadence is subject to aging it is necessary to conduct long term experiments. In the short time of the experiments for this work, certain tendencies have been

shown that give clues to a possible aging. Although it is to be assumed that the users know the text to type after the enrolment phase, i.e. on the same day, some features of typing cadence continue changing in the aftermath. The n-segment duration increased continually over the time of the experiment and the number of crossovers, on the other hand, decreased. The number of mistakes as well shows a downward trend. All these development have the consequence that the samples grow increasingly different from the reference pattern in time and the reference pattern is aging.

It is necessary to add that these developments are very slight and the features overall very stable. Nevertheless, it would be interesting to observe to what extent the feature families keep to this trend in a longer experiment and how strong is its influence on the identification quality. A further research topic is the question of how typing cadence reacts to old age, decreasing eye-hand coordination or arthritis.

The conclusion is that template aging is an important issue for typing cadence biometrics and its influence has to be considered in the plans of developing further architectures based on typing behaviour.

## **9 DESIGNING A FALL-BACK SOLUTION FOR A MULTI-FACTOR AUTHENTICATION USING BIOMETRICS**

---

Adding more factors to the authentication increases the security of a system. At the same time, the time and effort required on the side of the user in order to complete the procedure increase which results in a lower level of system usability. Beside that, using a more-factor authentication requires also the design of a multi-factor fall-back method. This chapter proposes a fall-back with combination of password, biometric and token.

---

### **9.1 Multiple factor authentication**

As mentioned before in this work, password based authentication has reached its limits, as it is not reasonable to ask common users to have always different passwords for every systems they use, to change these passwords every  $n$  days (where  $n$  is a number that is continuously decreasing) and most of all, to remember all these passwords and not to write them down. (BSI 2008) Also, the speed of brute-force attacks on passwords has increased (Breese 2007) so that the length of a secure password also has to be increased.

An alternative is to add more factors to the authentication, for example a biometric and a possession feature, thus achieving a multi-factor authentication. For this, we must consider the following influence factors:

- The desired level of security is in reverse correlation with the level of comfort. Upon using more-factor authentication, the AAI will be very secure, but also hard to access even for the entitled user.
- For every new authentication factor added, we must also supply a fall-back method in case the new factor fails (the password can be forgotten, the token can be lost or the biometrics can refuse to recognize the entitled user). At the design of the fall-back mechanism it must be considered that

the level of security provided by the fall-back should be equal to the one provided by the normal authentication procedure.

- If possible, the user should not be forced to carry extra authentication devices which can be forgotten or stolen.

- The individual level of security provided by each authentication mechanism is not considered here. We assume that the same level of security is offered by any of the factors used. Otherwise, it would be impossible to make for example a password fall-back using a token if we would consider the password more secure than the token itself. While possible, this prioritisation of fall-back methods does not play a role in our model.

The multiple factor authentication is also a requirement for the prototypes based on this work. The access to data in a system should always be granted if the user fulfils two authentication attributes. The methods used were Psylock, the biometric authentication method based on typing behaviour recognition and the widespread knowledge based combination of user name and password.

A fall-back solution to this combination of factors has to implement a mechanism that will prevent the impossibility of authentication using these two methods (biometrics or password). Therefore a knowledge based solution has been selected which is based on asymmetric cryptography and which still permits the use of the system in case of a loss of password or the impossibility of an authentication through biometrics.

## 9.2 Key management

To enable multiple factor authentication, a key management has been developed. In this model, it is intended that the user gives his user name and password upon registration. If the user name and password are accepted, a *salt* for hashing the password is generated and the hash function is used with the salt on the password. Subsequently the hash value, the *salt* and the user name are saved in a database. Furthermore, an asymmetric key pair consisting of a Public Key and a Private Key is generated. The user's password is encrypted via Public Key and stored in the database. The Private Key is offered to the user for download.

Unlike the previously mentioned security attributes for access control, the Data Encryption Key is responsible for the safe storage of data in the database. This symmetric key is randomly generated for each user at his registration but not saved as plain text in the database. Otherwise the data of the user could be decrypted using the key when the database is accessed. Instead of this, a

Password Based Encryption Key (PBE) is generated from this password with which the Data Encryption Key is encrypted. Here, it must be pointed out that the user's password is never saved as plain text in the database. This mechanism is presented in the following scheme:

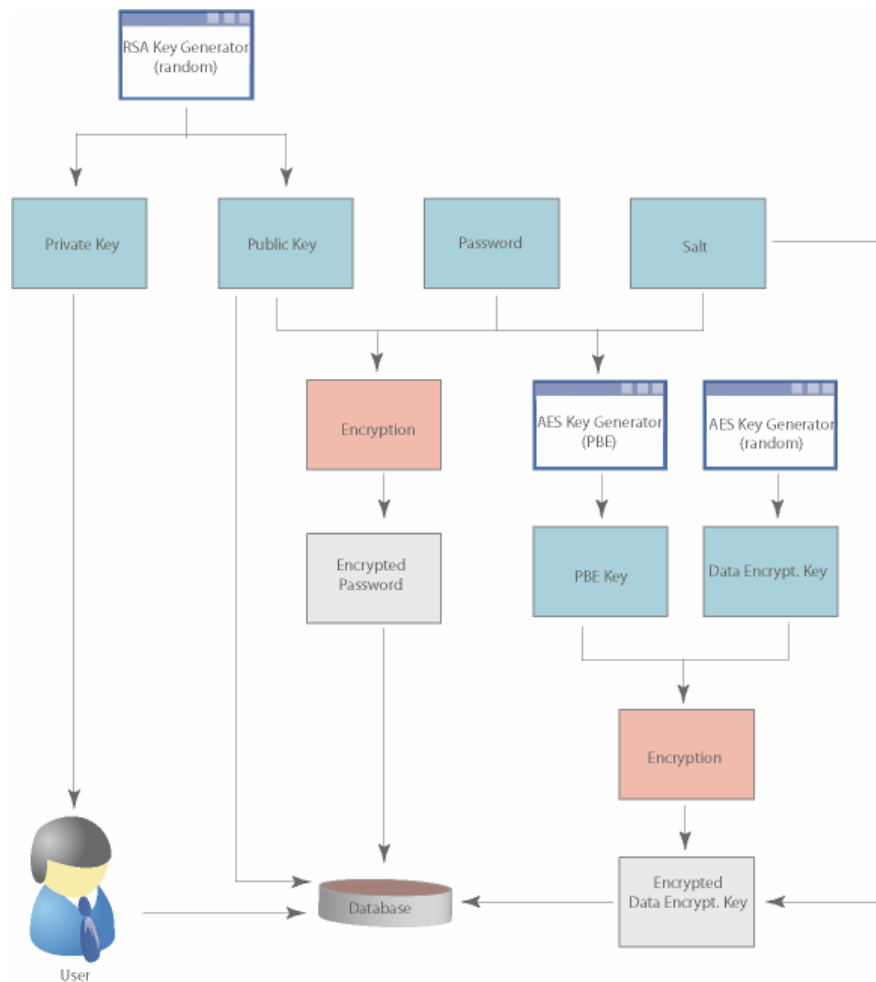


Fig. 9-1 Key management – Generation and storage of keys

Upon the login procedure, the comparison of the password hashes with the password submitted by the user takes place by using the hashing function. Further, the PBE Key is generated from the password and the Data Encryption Key is decrypted and saved in the user's session object.

In case of a fall-back, the user has to authenticate via password or biometrics and upload the Private Key he received upon registration. In case of a password fall-back, the forgotten password can be decrypted after the check of the Private Key, with which the Data Encryption Key can be decrypted and a new password will be issued. In case of a biometric fall-back, a new biometric enrolment is started and a new biometric profile will be created.

### 9.3 Fall-back mechanism

The fall-back mechanism has two components:

The first takes effect already during the enrolment process. At this time, an asymmetric key pair is generated for every user in the course of his registration. The Private Key is stored by the user at a safe location while the Public Key is deposited in the database. The user name and password are encrypted using the Public Key and also saved in the database.

These enrolment steps are indicated in the following table, which also considers whether the operations take place on the client or on the server side:

Client	Server
Username u Password x	
➔ Send u, x	
	1. Generate $salt = \gamma_u$
	2. Hash $h(\gamma_u, x) = h_u$ (PBE Key)
	3. Create asym. key pair $\langle P_u, S_u \rangle$ $P_u$ = public key $S_u$ = private key
	4. Encrypt password with public key $enc_{P_u}(X) = e_u$
	5. Generate Data Encryption Key: Random $r'$
	6. PBE Key: Encrypt $enc_{\gamma_u}(r') = e'_u$
	Save in DB: $\gamma_u, h_u, P_u, e_u, e'_u$ .
	➔ Send private key $S_u$ to client
7. Biometric enrolment Record 30 samples: S1, S2... S30	
➔ Send S1, S2... S30	
	7. Biometric enrolment: Encrypt all samples using Data Encryption Key $r'$ : $Enc_{r'}(S1), Enc_{r'}(S2), Enc_{r'}(S30)$
	8. Create biometrical template: Template(S1, S2... S30) = $\Lambda_u$ Encrypt template $enc_{r'}(\Lambda_u)$
	Goal: encrypted data $enc_{r'}(d_i) = e_i$

Table 9-1 Biometric enrolment with fall-back option

The second component is run during the actual fall-back phase. Then, if the user has forgotten his password or isn't able to authenticate biometrically he can upload his Private Key. Its authenticity is checked by the system by decrypting the encrypted user name from the database with the Private

Key and comparing it to the current user name. In case of a successful comparison, the user has to authenticate further by means of a second factor (password or biometrics) and, in the end, has the fall-back option to reset his password or re-launch the biometrics enrolment. In the following will be described an example of the procedures a user and a server site have to go through in case a password is forgotten.

Client	Server
Username u	
Password x	
➔ Send u, x	
	1. Get $\gamma_u, e_u$ from database.
	2. Hash $h(\gamma_u, x) = h_u$
	3. $dec_{h_u}(e_u) = r'$
4. Get sample $S_j$	
➔ Send $S_j$	
	5. $S_j \parallel \Lambda_u = \text{pwd}$

Table 9-2 Biometric authentication with fall-back option

Firstly, a user indicates that he has forgotten his password by clicking the according link („Lost your password“). In the next step he is asked to enter his user name and to upload his Private Key. Its authenticity is checked by the server and the user is forwarded to the biometric authentication in case of success. After a successful biometric authentication, the user logs in and receives the possibility to change his password. After he has chosen a new password, following steps are taken by the system:

- The old user password which is stored encrypted in the database is decrypted using the Private Key.
- The Data Encryption Key is decrypted using the PBE on the base of the old password.
- The Data Encryption Key is re-encrypted using the PBE on the base of the new password and stored again in the database.
- The new password is encrypted using the Public Key from the database.
- A new hash value is generated for the new password (for the login).
- The hash value, the encrypted password and the encrypted Data Encryption Key are saved in the database.

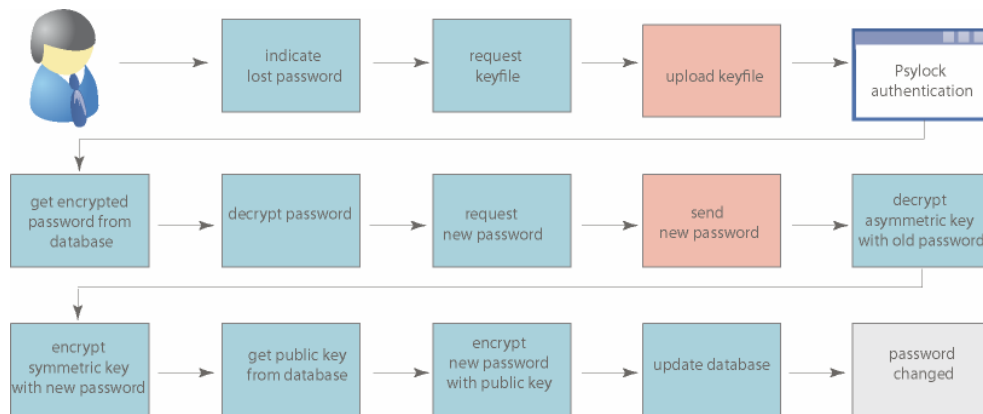


Fig. 9-2 Fall-back mechanism in case of a forgotten password

## 9.4 Fall-back problems

The fall-back algorithm provided here assumes that two factors are used for the authentication, while a third one is available for the fall-back solution. As we do not want to make the fall-back with only one factor for security reasons, the solution presented here uses also two factors for the fall-back phase, which means that a user can either forget his password and be recognized by the key file and the biometric method, or the user is not recognized by the biometric but can authenticate with the key file and the password. This restriction can be lifted either if we consider that the key file offers enough security for the fall-back phase or upon the use of a fourth mechanism (which can be another biometric or a call to a help-desk and the communication of some personal data).

If we were also to prioritise the security level of each form of authentication factor, we could make the following assumption: the password used for authentication can be reset only with another password (like the answer to a knowledge question), the biometric method can be reset only by another biometric method (like fingerprint or iris scan) and the key file can be reset only with another token (software or hardware). These procedures are more complicated than the one proposed in this chapter and were not considered.

A better security of the system could be achieved if the biometric method itself could be used to generate keys that would encrypt or decrypt data. As biometrics change continuously, biometric key generation is very hard to achieve. Attempts in this direction were made by (Kholmatov 2006) who was able to obtain keys up to 128 bits from signature-based biometrics.

## 9.5 Conclusion

As seen in this chapter, fall-back is possible even when working with multiple factor authentication. However, its complexity increases with the number of factors used. The solution proposed here is based on the fact that the user has three factors available of which he can use two for authentication. In the case that both password and biometrics are unavailable, another (manual) solution should be used, such as the direct contact with an administrator.

For even higher security, the third factor (key file) can be also used for authentication, thus achieving a full three factors authentication (knowledge, possession and being). As the biometric AAI model that was the aim of this work has been comfortable for the user, using a key file that the user must always have available was not regarded as necessary.

### **10 BIOMETRIC AAIS WITH SYNCHRONISED DATA**

---

Chapter 5 about biometric AAIs presented possible biometrics problems which were investigated for the example of typing behaviour biometrics. The results of this analysis lead to the conclusion that synchronising biometric data in a federation would decrease these risks. This chapter discusses the design of a biometric AAI using synchronisation. Use-cases show which mechanisms could be used in order to synchronise data and to create a biometric system that is actual, highly qualitative and resistant to replay attacks.

---

#### **10.1 Introduction**

##### **10.1.1 Combination of biometric methods with AAIs**

The basic idea of OpenID is the outsourcing of the login process of participating web applications, so called consumers, to dedicated identity providers managing authentication and profile data. As a consumer does not execute the authentication process anymore, the responsibility for safety and availability of user data is outsourced to the identity provider. In many cases this is desirable, as the introduction of identity providers causes a specialization on authentication services. Identity providers regard the safe management of authentication data as a business model and not as a necessary evil, which leads to a general increase of safety.

The downside of this development is that a consumer becomes dependent of identity providers as he is not able to authenticate users anymore. In order to diminish this dependency, many consumers decide to execute an own authentication beside the support of an AAI. A redundant biometric authentication mechanism may be secure, but it negatively influences the user friendliness of a system, as a user has to enrol separately to every system.

The ideal solution is the use of a Circle of Trust for the transfer of biometric authentication data. In this case, a user would only have to create a biometric profile once and use it for other applications in the Circle of Trust. These applications can then authenticate users independently as they are in possession of the user's biometric profile.

## **10.2 Problems and requirements of a Circle of Trust**

In the following, the problems and requirements of the realization of a Circle of Trust with a synchronised biometric account will be discussed on a general level independently on specific systems.

### **10.2.1 Single Sign On**

A user friendly function would be the use of the CoT as a Single Sign On. The user signs up at a server in a network and later does not have to authenticate at another server to gain access. He receives a ticket valid at all other servers in the CoT and is accepted for authentication by means of this ticket.

### **10.2.2 Attribute management**

If several applications act independently in the Circle of Trust, they store not only the access data but also user attributes, e.g. the address, on their own servers. As this leads to redundancy, it is necessary to conceive mechanisms of adequately storing and sharing this data.

One possibility is the outsourcing of attributes to a central server. The respective applications then contact the central server on demand, which brings the advantage that no redundancy can occur. For data privacy protection reasons, the user is asked to confirm whether an application should be granted access to certain attributes. A disadvantage of this mechanism is the additional intermediate step in authentication, which has a negative influence on system performance. A possible compromise is a one-time contact to the server and the later storage of attributes by the application. The problem hereby is the future actualization or synchronisation of attributes.

To avoid the need for a central instance, all attributes can be mirrored. A necessary decision is to define whether the mirroring is centred at one main server as a master-slave solution or whether the actualisation of attributes can be executed by every application.

### **10.2.3 Assignment of user names**

#### **10.2.3.1 User names valid for the entire Circle of Trust**

The easiest solution is storing the same user name in all applications in the Circle of Trust. Upon login or synchronisation, biometric data is transmitted to another server without the previous mapping or assignment of different user accounts. On the other hand, every application will have to store all possible user names registered in the CoT, which is unnecessary if the user only accesses a few applications from the circle.

This solution can be realised in two ways:

1. The user name is set centrally by the administrator and stored in the databases of the separate applications. This leads to questions about where the initial enrolment process has to take place. One possibility is that it is conducted on any server in the CoT and the account and biometric data is transmitted to the other servers. Another way is to use a specially designated server for the enrolment.
2. When a user creates a new account at a provider, he is subsequently registered to all other applications. After enrolment, the biometric data is transmitted throughout the CoT. Upon registration it is necessary to check whether such a user name is already assigned in the Circle of Trust. This can be done by a central server that can give information about all assigned user names. Alternatively, all servers in the Circle of Trust can be asked whether the user is already registered at them. Ideally, only one request is necessary if all user databases are synchronised.

#### **10.2.3.2 Individual user names for every application**

A further possibility is that a user can have different user names for every application. An advantage of this alternative is that there is no unnecessary storage of user names at every provider. The user John can be registered at providers 1 and 2 but not at provider 3 so that the user name John can be collocated to another user there. This solution has to specify the way in which a user name is assigned to other user names at different providers. There are two scenarios for the mapping of user names.

##### **10.2.3.2.1 Use of a mapping table**

In this case, a table shows the different user names of a person on the servers in the Circle of Trust. It is necessary to define where such a table should be located:

1. There is one central server responsible for the mapping of user names:

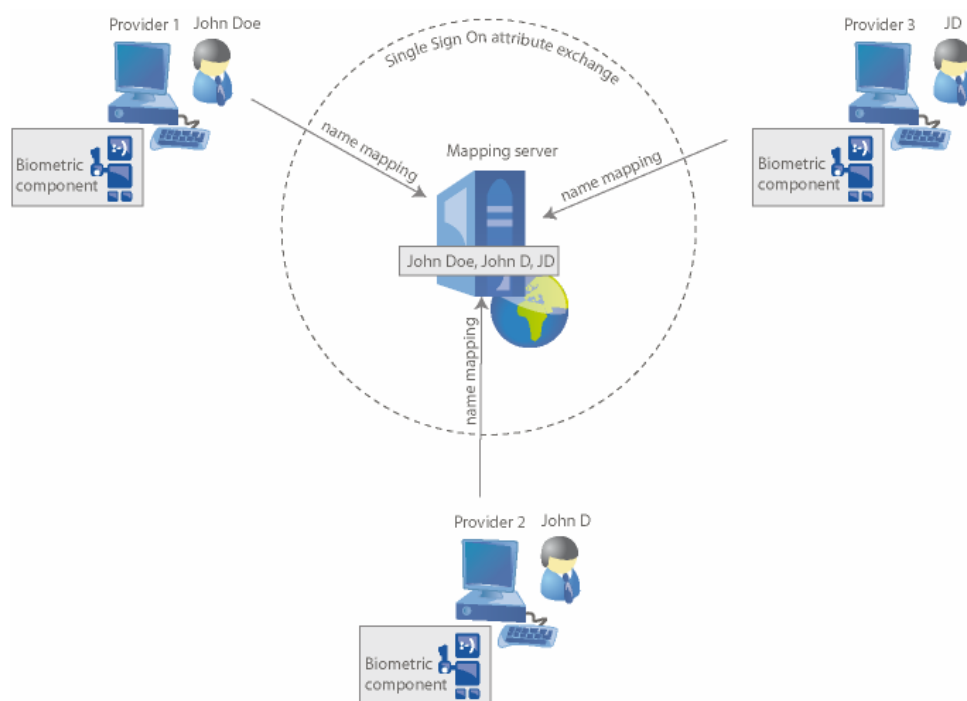


Fig. 10-1 Use of a central mapping table

**Advantages:** As such a table is located on a single server, this solution is the least redundant model. It also meets the users' need for data privacy, as it is not possible for the providers to verify the correlation between the profiles and to retrieve information about the user against his will.

**Disadvantages:** The central server must be contacted during the synchronisation in order to find the corresponding user account on other servers. If this server is offline, the synchronisation is not possible, and a "single point of failure" occurs.

2. The mapping table is located at every provider in the CoT.

**Advantages:** The central mapping-server is no more necessary as an intermediate instance, which results in less complexity and higher data availability.

**Disadvantages:** This solution creates more overhead and provides no protection of data privacy, as every provider has information about all the different user names of a single person in the circle of trust. This architecture makes different user names obsolete.

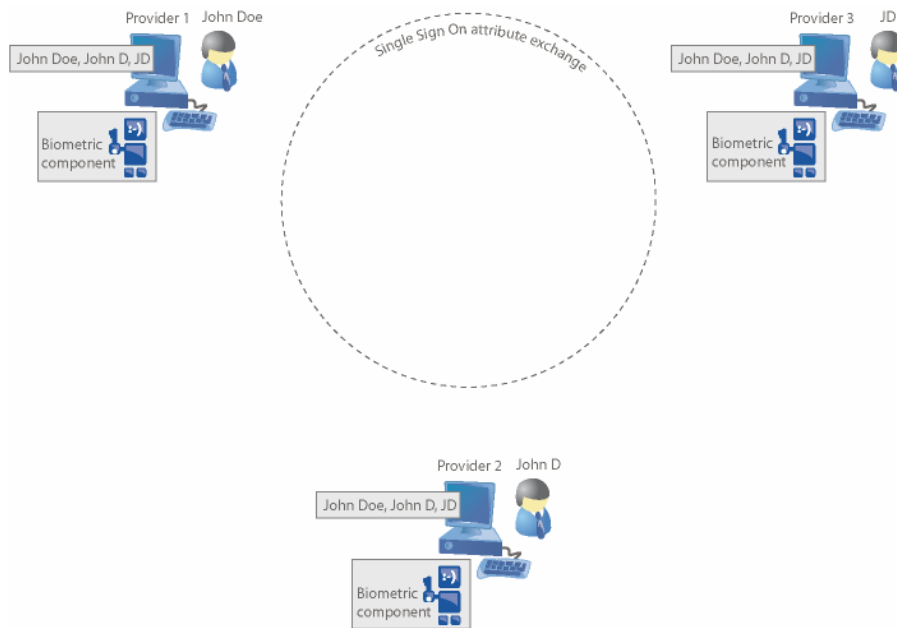


Fig. 10-2 Mapping table stored by each IdP in the circle of trust

Regardless of the fact whether such a table is centrally stored or saved as a copy at each provider, the important issue is the connection between user accounts. In a user-centric approach, the user can specify which other accounts he possesses at other providers in the circle. Another possibility is to use biometric authentication in order to find identical biometric profiles and thus to map the corresponding user names.

#### 10.2.3.2.2 Dynamic assignment of accounts by means of biometrics

A different approach is to forego mapping tables and to localize the user on different servers by a biometric sample:

The user registers at IdP1 via biometrics. The biometric sample is sent to other servers such as IdP2. IdP2 now matches the received sample against all profiles stored locally. If the achieved match score is higher than a certain threshold, e.g. 70%, IdP2 assumes that the current sample belongs to one of the registered users. The user can now log in to IdP2 and the biometric sample is added to the identified profile for synchronisation purposes.

Advantages: This solution saves disk space and increases availability as it requires no update of a mapping table and to contact to dedicated servers. User privacy is protected by the fact that nowhere does it become apparent under which user name a user is registered at other IdPs.

Disadvantages: This variant requires a high computational effort. In the worst case, all profiles on the server have to be matched against a biometric sample. This worst case is not improbable; it occurs always when a user does not have an account on a server and therefore no matching profile can be found.

The worst case scenario follows the formula:

No. of necessary typing example comparisons = (No. of servers in CoT -1) \* (No. of profiles per server)

Another problem occurs when the achieved match score is too small to ensure a clear biometric identification. In this case the synchronisation can not be executed automatically, but only by means of manual mapping by the administrator or by the user himself.

## 10.2.4 Mirroring of biometric accounts on the example of Psylock

### 10.2.4.1 Psylock data to transfer

If a user registers to an application in the Circle of Trust while already having a Psylock account elsewhere in the CoT he has to have the possibility to transfer the complete profile to other applications.

If the user logs in to an IdP where his Psylock account has already been created, the new typing sample has to be shared with the other participants in the Circle of Trust. This has the purpose of synchronising the Psylock profiles of a user in real-time. Real-time synchronisation is important, amongst others, for the protection from replay attacks discussed in the next section.

Simplified database structure for the storage of Psylock profiles:

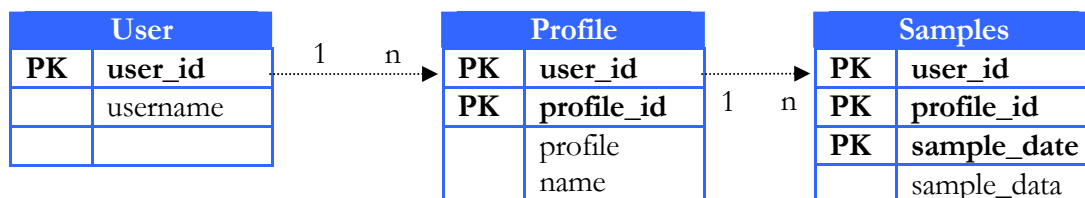


Fig. 10-3 Simplified biometric database structure

With this structure, a user can create separate profiles for different keyboards. This procedure is especially useful for very different keyboards such as notebook, desktop and ergonomic keyboards. A single typing example is build after the following schema:

„2007-11-26\_13:10:53&066v9999&065v0031&066^0016&065^0047&067v0156&067^0031”

The following table shows a simplified construction of a biometric database:

username	profile	sample_date	sample_data
John	Notebook	2008-11-26_13:12:29	066v9999&065v0031&066^0016&065^0047&067v015656&067^0031
John	Notebook	2008-11-28_19:10:41	066v9999&065v0031&066^0016&065^0047&067v015656&067^0031
John	Institute	2008-11-21_11:40:36	066v9999&065v0031&066^0016&065^0047&067v015656&067^0031
Matthias	Institute	2008-11-24_22:19:12	066v9999&065v0031&066^0016&065^0047&067v015656&067^0031

Table 10-1 Biometric database

Synchronisation demands that profile data be transferred via a network or the internet. Therefore, it is important to calculate the emerging amount of data:

Date: 19 characters	= 19 Bytes (ASCII)	
One event: separation mark "&"		= 1 Byte
+ Key code (e.g. 066)		= 3 Bytes
+ Key down "v" or key up "^"		= 1 Byte
+ Time in ms (z.B.0012)		= 4 Bytes
		= 9 Bytes per event
One key: 2 events (key up and key down)		= 18 Bytes per key

Required memory for the sentence: “Ich bin der Meinung, die richtige Antwort lautet.”:

19 Bytes (Date) + 49 (Number letters) \* 18 Bytes (Memory per)  
= 901 Bytes per typing sample

For one profile: 9 sets = 9 \* 901 = 8,109 Kilobytes

This calculation does not yet incorporate the profile name which has to be transmitted as additional information. With a profile name of up to 20 characters it amounts to 901 + 20 = 921 Bytes per typing sample.

According to that it, the amount of data is 9 \* 921 = 8,289 Kilobytes per profile.

Therefore, the transmission volume for 100 users with 2 profiles each using the sentence “Ich bin der Meinung, die richtige Antwort lautet.” is:

100 users \* 2 profiles \* 8,289 Kilobytes per profile = 1,657,800 Megabytes.

This calculation does not include the space necessary for user names but assumedly it does not have to be sent additionally in an attribute exchange within a Circle of Trust.

#### **10.2.4.2 Necessary actuality due to replay attacks**

A replay attack on typing behaviour biometrics can look as follows: An attacker intercepts a typing sample of Alice using a key logger. He saves not only the sequence of keys but also parameters relevant for typing cadence such as time intervals between key press and key release events. With such a tool, the intercepted typing sample can be reproduced in the exact same way as if Alice typed it in person.

An intercepted and unaltered typing sample would be recognized as a replay attack as Psylock assumes that such a high resemblance of two typing samples in the range of milliseconds can not be achieved by a manual login of the user. Even if the attacker replays the intercepted sample in a slightly modified way, the biometric method is able to detect this modified replay attack using the methods presented in this work.

In the context of a Circle of Trust with multiple synchronised biometric accounts, a successful detection of replay attacks makes it necessary that user accounts are synchronised in real-time. If this requirement is not met, the following scenario is probable: An attacker intercepts a typing example of Alice at IdP1. Alice also has an account at IdP2 which is not regularly synchronised with IdP1. Should the attacker replay the IdP1 typing sample to IdP2, the latter is probably unable to detect the replay attack as an unnaturally high match score can not occur due to an outdated profile of Alice.

#### **10.2.4.3 Synchronisation failures**

An obvious problem with the synchronisation of biometric data occurs when the transfer of a typing sample or a whole profile fails, for example if an IdP is offline.

To avoid this case, the IdP that received the latest typing sample and started the synchronisation process must remember with which other IdP the synchronisation failed and retry at a later date.

Another solution is for the server which was not accessible to inquire whether new typing samples were delivered at the other participants. As additional information he sends the date of his latest sample.

### 10.3 Synchronisation on the database level

Synchronisation can be conducted at different levels. One possibility is to regard biometric account data as common attributes and therefore to transmit them via the protocol of the respective AAI.

Disadvantage: Synchronisation taking place on a high protocol level is not high-performance, as opposed to database synchronisation. Also, a synchronisation protocol would have to be written considering various scenarios of synchronisation failures between IdPs.

An efficient possibility is the synchronisation of biometric accounts data directly on the database level; this uncouples it from the overlying AAI. The AAI remains responsible of authentication and authorization, Single Sign On and the exchange of remaining attributes. An advantage of this procedure is a better performance and a lower implementation effort as there already are various software solutions for the mirroring of databases. (Qarchive 2008) On the other hand, most of these solutions presume a master-slave relationship between the database servers, which is not the case in an emancipated Circle of Trust model.

Basically, there are two possible scenarios for database synchronisation:

1. Biometric data can be managed via a central repository:

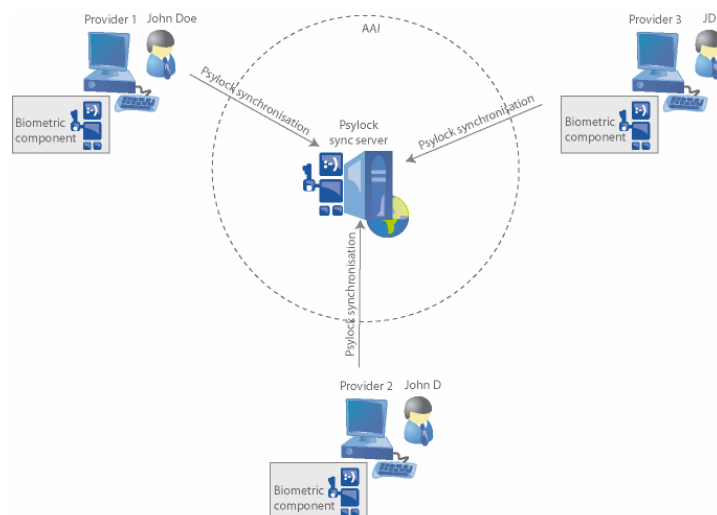


Fig. 10-4 Central repository

When a user logs in to IdP1, this latter commits the new sample to the synchronisation server that updates IdP2 and IdP3 either automatically or upon request. This can be realised by merge replication with a central distributor.

The former variant creates a Single Point of Failure. An alternative is a decentralized configuration:

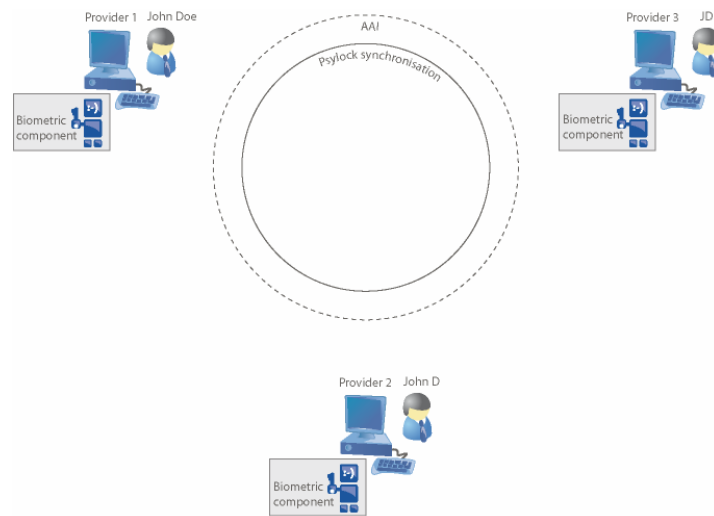


Fig. 10-5 The decentralized version

However, there are no replication mechanisms allowing a completely decentralized synchronisation of multiple emancipated servers.

Generally, the mirroring of biometric data on the database level is an interesting possibility due to the high performance. However, this solution implies considerable restrictions: A close relationship of trust between the participating partners is necessary, as the IdPs in the Circle of Trust have to grant other IdPs access to their user databases. A further requirement for the synchronisation on database level is that user data has to be identical at all IdPs, meaning identical user names on all servers in the CoT. This makes it impossible for the user to assume different user names, e.g. for the protection of privacy. It also automatically creates user accounts at all IdPs in the CoT, although the user may want to use only some of them. This architecture allows a loose link between IdPs in the CoT; unlike in the AAI, the user has no influence upon which IdPs are allowed to access his profile data. Due to these restrictions, profile data is treated as attributes and synchronised on the AAI level in this work.

#### 10.4 OpenID Attribute Exchange Extension

The Attribute Exchange Extension in OpenID can be seen as the successor of the Simple Registration Extension. Its function is to transmit attributes from the identity provider to the consumer. The structure of the Attribute Exchange Extension is not much different from its forerunner: instead of complex XML structures, only keyword pairs are transmitted, e.g. "nickname=matthias".

One innovation is that the number of attributes is not limited in advance. Instead, it is possible to add new attributes and to name them explicitly using URIs. In order for attributes to be supported by as many consumers as possible, a standardizing process of commonly used attribute types is currently in progress (axschema.org, 2008).

An even more important innovation has been achieved by the possibility of transmitting attributes not only from the identity provider to the consumer but also in the opposite direction. This way, a consumer can request that certain attributes be stored at the identity provider, which can take place after a confirmation by the user.

Furthermore, a consumer can request that an identity provider automatically sends updates to the consumer if certain attributes have changed. By stating an update URL, a consumer can subscribe to attributes.

The main types of requests handled by the Attribute Exchange Extension are:

- Fetch Requests, where a consumer requests attributes from the identity provider;
- Store Requests, where the consumer saves attributes at the identity provider;
- Asynchronous updates, where an IdP updates attributes at the consumer without interaction with the user.

These requests are sent during a normal authentication request making possible the following situations:

- The IdP can decide based on a predefined policy which attributes are sent to the customer. For example, the user may define that different E-mail addresses are sent to different consumers.
- The IdP can ask for the users approval for the attributes sent to the consumer.
- The store request can also be made with the explicit confirmation of the user.

The parameters that can be transmitted by the Attribute Exchange Extension (AX) can be grouped as follows:

#### A. Fetch Request:

In this case, more fields will be added to the authentication request:

- Openid.ax.mode: contains information about the used AX mode, in this case “fetch\_request”.
- Openid.ax.if\_available: a comma-separated list of attributes that the consumer can receive.
- Openid.ax.required: a list of attributes necessarily needed by the relying party (RP); this list is mandatory.
- Openid.ax.type.alias: the URI uniquely defining the attribute.
- Openid.ax.count.alias: the number of values that the relying party wants to receive as attribute.
- Openid.ax.update\_url: the URL of the consumer that will receive the asynchronous attribute update.

In order to define unique attribute types, Uniform Resource Identifiers (URIs) are used.

An example for the authentication request with defined biometric attributes along with standard attributes like e-mail, address and name:

```
openid.ns.ax=http://openid.net/srv/ax/1.0
openid.ax.mode=fetch_request
openid.ax.required=name, profilename, sampledata, sampledate
openid.ax.if_available=email
openid.ax.type.name=http://axschema.org/namePerson
openid.ax.type.email=http://axschema.org/contact/email
openid.ax.type.profilename=http://psylock.com/profilename
openid.ax.type.sampledata=http://psylock.com/sampledata
openid.ax.type.sampledate=http://psylock.com/sampledate
```

When the authentication is successfully conducted at the level of the identity provider that supports the attribute exchange extension, this IdP can send as answer the following fields:

- Openid.ax.mode: contains the “fetch\_response”
- Openid.ax.type.alias: The URI for the attribute.
- Openid.ax.count.alias: The number of values that have to be returned for that attribute.
- Openid.ax.value.alias: The value of the attribute when the “count” parameter is not set.
- Openid.ax.value.alias.n: The attribute number “n”, when the “count” parameter is set.
- Openid.ax.update\_url: The update URL for asynchronous updates.

The answer to the previous requests can have the following syntax:

```
openid.ns.ax=http://openid.net/srv/ax/1.0
openid.ax.mode=fetch_response
openid.ax.type.name=http://axschema.org/namePerson
openid.ax.type.email=http://axschema.org/contact/email
openid.ax.type.profilename=http://psylock.com/profilename
openid.ax.type.sampledata=http://psylock.com/sampledata
openid.ax.type.sampledate=http://psylock.com/sampledate
openid.ax.value.name=John Doe
openid.ax.value.profilename=Laptop
openid.ax.value.sampledata=&066v9999&065v0031&066^0016&065^0047&067v0
156&067^0031
openid.ax.value.sampledate=2008-10-22_12:19:41
openid.ax.count.email=0
```

In this answer, the “email” parameter did not return any answer, as it was optional.

#### B. Store Request:

The store request is used to submit data from a consumer to an identity provider. The format of the store request is almost the same as the fetch response. The only difference lies in the parameter “openid.ax.mode” which can be set on the value “store\_request”. Upon the completion of a store request, the IdP sends back a store response, which contains only the field “openid.ax.mod”. This field can have one of two values: “store\_response\_success” or “store\_response\_failure” depending on whether the store action could be completed or not.

The store request is an important option as it allows a bidirectional exchange of attributes.

#### C: Asynchronous Updates:

A disadvantage of the simple registration extension is the fact that attributes can be transferred only during the login process. This restriction does not exist in case of the attribute exchange extension, as an IdP can send so-called “unsolicited responses” to the customer, which represent assertions that a customer did not request, but which are sent by the identity provider, e.g. whenever a certain attribute has changed. This assertion contains the attribute, similar to a fetch response.

It is also possible for the customer to inform the identity provider, for which attributes he wants to receive automatic updates. For this, the consumer sends a fetch request with the URL identifier of the user in the field “openid.as.update\_url”. In case of an asynchronous update, these changes will be initiated by the identity provider. As in the first phase the identity provider does not have any shared secret with the consumer, it will sign the assertion with its private key. In order to check the

signature, the consumer must contact the identity provider. This procedure corresponds to the “stateless” mode of OpenID which does not use a predefined share secret as signature.

## **10.5 Scenarios for a circle of trust with OpenID**

Originally, it was not foreseen for OpenID to be integrated in a structure like the circle of trust. The original purpose of OpenID was exactly the contrary: instead of using a predefined trust network, the dogma of OpenID stated that each standard service provider should be able to cooperate with every standard identity provider in the web. Even in the case when several dedicated identity providers are available, it is not foreseen that these should cooperate in order to verify, for example, online tickets emitted by another identity provider. The advantage for the user is the fact that he should not be forced to register with several providers in the web.

The attribute exchange was also poor in OpenID 1.0 in comparison to other AAI, as only few attributes could be sent using the “simple registration extension” and this could be done only from the identity provider to the consumer.

Nevertheless, this has changed since the introduction of OpenID 2.0: the “attribute exchange extension” provides a strong interface for the transmission of different attributes in both directions. Additional functionality like the request for different authentication methods or the possibility to implement a real Single Sign On have lead to the selection of this AAI as a solution for integration within a circle of trust.

This chapter will discuss different configurations of the integration of OpenID in a predefined trusted network.

### **10.5.1 1<sup>st</sup> configuration: one identity provider and more consumers**

The first use-case presumes that a user is registered to an identity provider IdP1. This identity provider uses biometrics as an authentication method. At the same time, the user accesses the consumers SP1 and SP2, which allow an authentication over IdP1. Additionally to that, SP1 and SP2 can provide their own local login procedure, also via biometrics. In this configuration, there is a third consumer SP3, whose function will be explained below.

In this case, both consumers are not fully dependent on the identity provider which prevents a single point of failure. This dependency exists only in the case of the consumer SP3. This dependency is desired, as this consumer should be as light-weighted as possible and therefore does not require its own authentication.

This scenario can be seen in the following graphic:

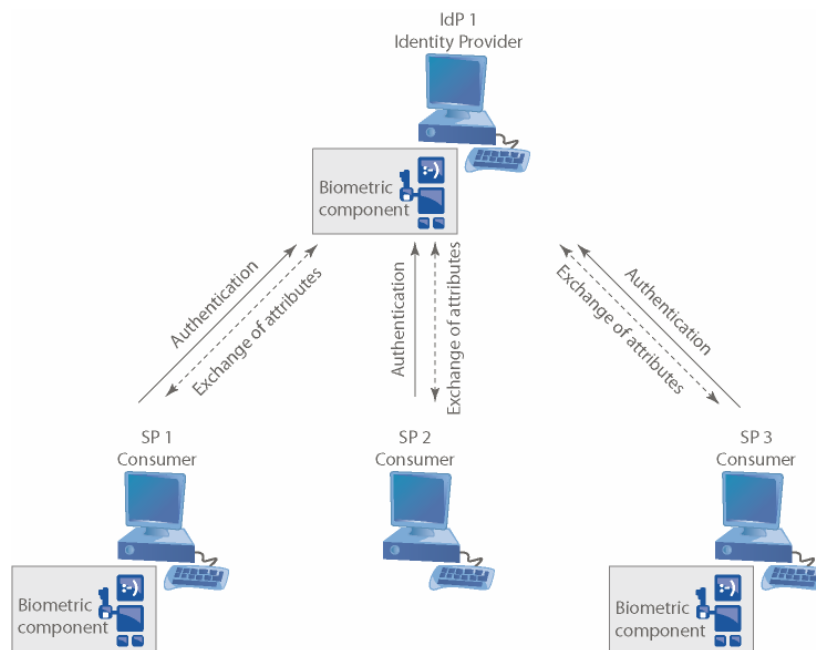


Fig. 10-6 First configuration

This configuration brings several advantages to the user:

1. The user can use a partial Single Sign On with OpenID: upon login to his identity provider IdP1, he has to provide only his OpenID URL to the other customers in order to be automatically authenticated.
2. The user can use a full Single Sign On, in the case that these components work also with unsolicited assertions. The user logs in to IdP1 and has the option to remain logged in with all other consumers. This uses the “stateless” mode of OpenID as the identity provider does not know which shared secret must be used for data encryption. For this, it has to encrypt the message with its private key.
3. The user can use a strong biometric authentication even for the light-weighted consumer SP3. As mentioned before, biometric authentication can have different types of problems when used in circle of trust structures. It is therefore necessary to synchronise biometric data between the components of the circle in order to have a secure and user-friendly system. For this, a communication must be established. This always leads to the identity provider, as there is no possibility to exchange information between the consumers SP1 and SP2. This is a potential disadvantage as the identity provider has the problem of a “single point of failure”. Other

alternatives are the use of a combination of the identity provider and the consumer or the possibility that a user has more identity providers. These scenarios will be presented later in this chapter.

In order to start the scenario, the user must register to the identity provider. As this process uses biometrics, the registration will also contain a biometric enrolment.

#### 10.5.1.1 Enrolment workflow

First use-case:

The user enrolls to IdP1 which then submits the profiles to the consumers. The following graphic shows the necessary workflow:

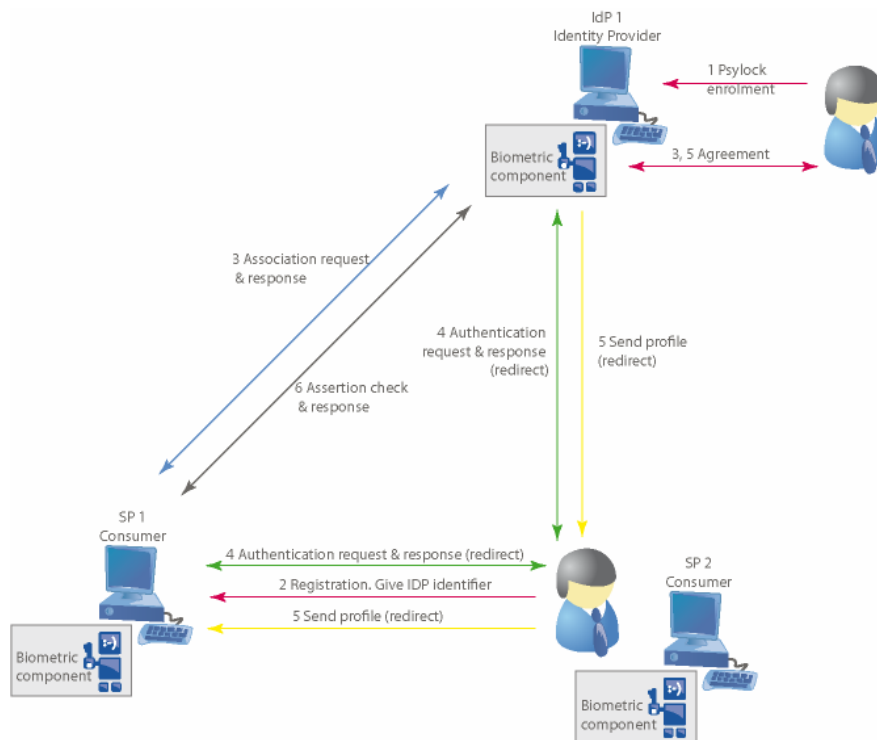


Fig. 10-7 Enrolment workflow

Enrolment steps:

1. The user registers to IdP1. The biometric enrolment process is completed and a biometric profile created.

2. Later, the user creates an account at SP2. For this, he does not have to enrol again, but types the URL of his identity provider.
3. The consumer SP2 contacts the identity provider and agrees upon a shared secret with IdP1, which will be used to sign the authentication messages (assertions). The user confirms this step.
4. The SP2 sends an authentication request to IdP1, which answers with a positive authentication response. The response can already contain the biometric profile of the user, if this is marked as “required” in the request.
5. In case no biometric profile was submitted in the authentication request, the IdP1 sends per Update URL an unsolicited response containing the biometric profile. This response is signed with its private key.
6. In order to check the authenticity of the message, the SP2 consumer contacts IdP1 which works in stateless mode.

Second use-case:

The user has a biometric profile at both consumer and IdP1. If the consumer works together with IdP1, these profiles must be synchronised as follows:

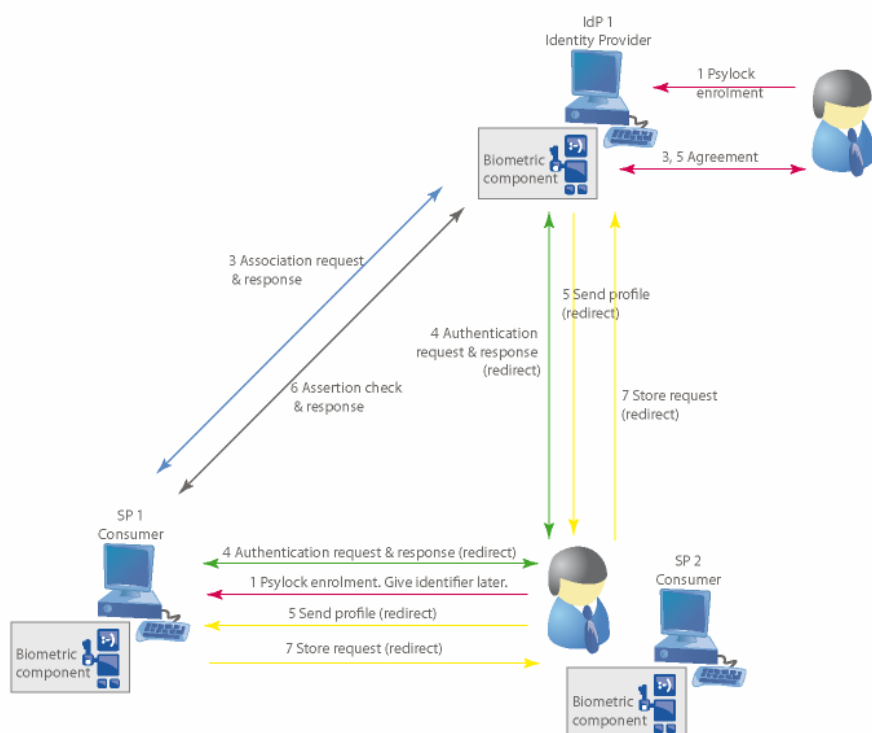


Fig. 10-8 Second use-case

Synchronisation steps:

The premise is that the user already has an account at IdP1 and SP2. Steps 1-4 remain the same as in the previous use-case. The remaining steps are:

5. IdP1 sends all profile data to SP2 in an unsolicited response. SP2 decides which biometric samples he will include in his profile by means of the date. The date of the biometric sample can be used as a key, as a user cannot “be” in two places at the same time, that is he cannot authenticate biometrically at the same time with the same sensor in two places. The only assumption here is the fact that the two parties are using synchronised times.

6. SP2 checks the authenticity of the unsolicited response by means of direct communication.

7. SP2 sends all his profile data to IdP1 in a store request. Here it will also be decided by means of the time stamp which of the biometric samples will be taken over into the database. IdP1 decrypts the information using the common shared key generated before.

8. For further synchronisation processes, the parties can first send the time of the last stored sample (for example, after every user login), so that the other party can know which samples to ask or to send.

Another interesting observation is the fact that, in order to synchronise more biometric profiles of the same user (e.g. if the user has more sensors), the profiles that were made with the same sensor must be also called the same by both parties. If this is not the case (e.g. the user has two profiles for the same sensor - „Notebook” and „Laptop”), the synchronisation process will result in two redundant profiles stored on each server. While the biometric method itself can detect that the two profiles with different names actually belong to the same sensor, it is better to leave the option of renaming profiles to the user.

After the enrolment phase is over, the user has biometric accounts at IdP1, SP1 and SP2. If he wants to login in to any of these parties, the newly generated sample will be immediately sent to the other parties from the circle. For this, there are another two possibilities:

### 10.5.1.2 Biometric login at the IdP

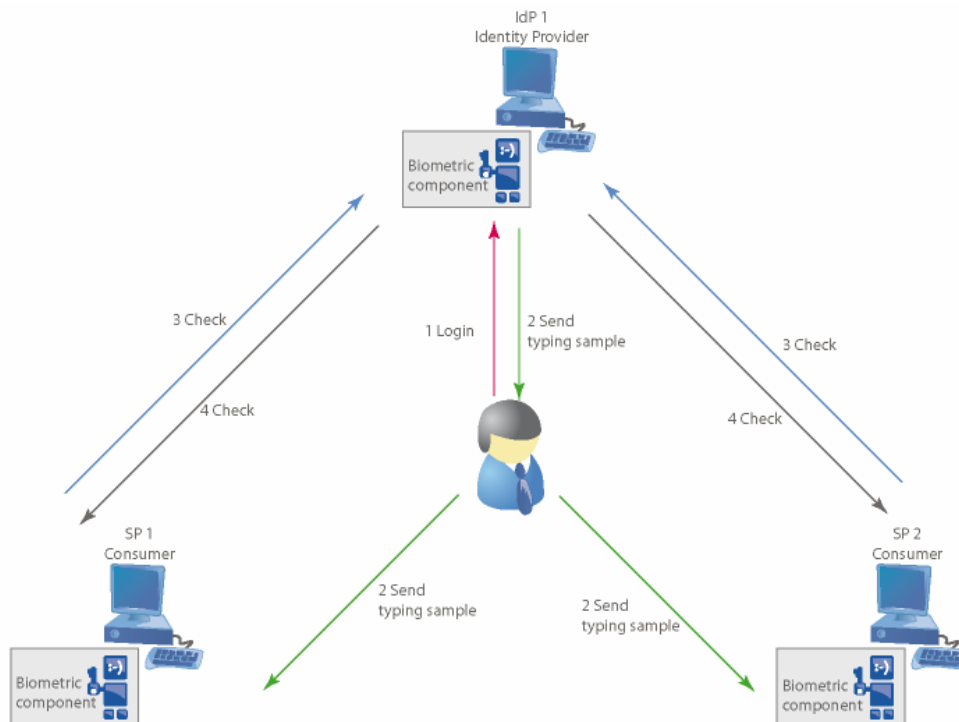


Fig. 10-9 Biometric login at the IdP

Steps:

1. The user logs in to IdP1.
2. IdP1 uses the attribute exchange extension to send the new biometric sample to the consumers by means of an unsolicited authentication response.
3. The consumers check the authenticity of the signature used in the unsolicited response.
4. IdP1 confirms the authenticity. The consumer can now save the sample.

### 10.5.1.3 Biometric login at the consumers

If the user logs in first at the consumer, which has to send a store request to the IdP in order to submit the new biometric sample. However, such a store request is part of an authentication request, which implies that the user must login first to IdP1 or that he already has an active session there. This procedure is followed for security reasons, as otherwise it would be possible for somebody to send a fake store request to the IdP.

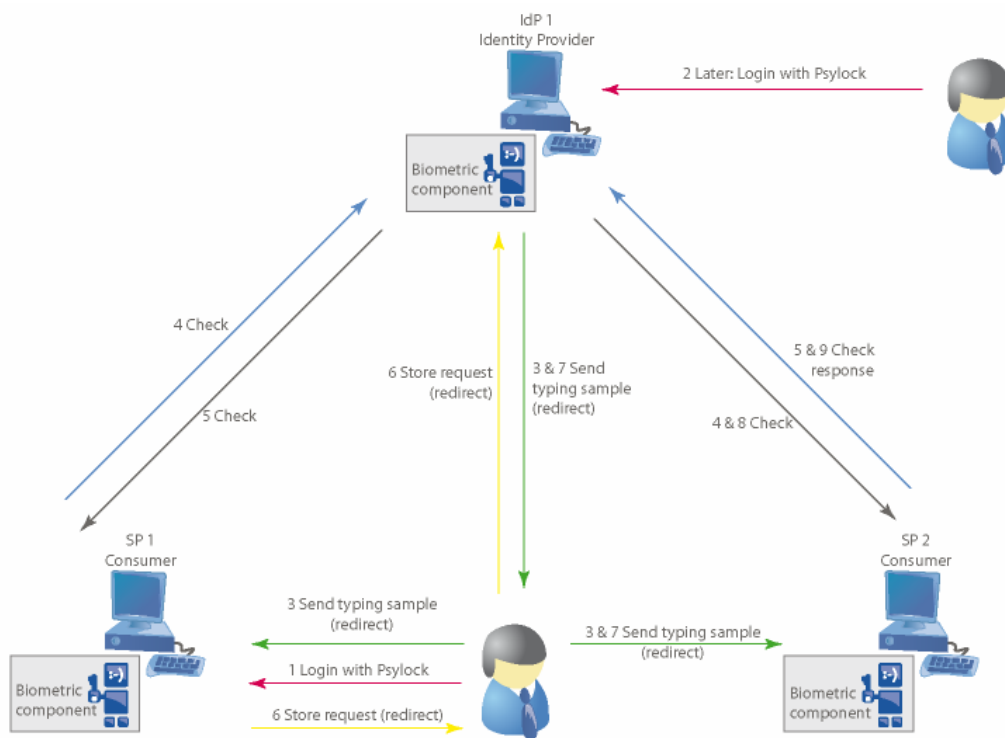


Fig. 10-10 Biometric login at the consumers

The process of re-login to IdP1 is not reasonable for the user, as he just logged in to the consumer. Due to this, an immediate synchronisation of the account cannot be made.

When later the user authenticates to IdP1, the logic flow is like in the steps 3-5 of the first scenario: the SP1 and SP2 are updated per unsolicited response. The difference here lies in the fact that consumer SP2 has already marked that he has to send the last sample to the IdP. As IdP1 also sends a biometric sample to this consumer, SP2 knows that the user has logged in to this IdP.

The next steps of this process are:

6. SP2 sends a store request to the IdP 1 using HTTP redirect.
7. After the IdP has processed the store request, it sends the new sample to other consumers as well in order to update them. This is done like in the first scenario using the update URL. The only difference is the fact that SP2 will not be re-updated, as the IdP knows that it received the new typing sample from SP2. This prevents an endless loop between IdP1 and SP2.
8. The consumers check the authenticity of the response by means of direct communication.

### 10.5.2 2<sup>nd</sup> configuration: a server is used as consumer or as IdP

In the previous configuration, a clear distinction was made between the roles of identity provider and consumer. This led to the fact that the attribute exchange was only possible over the identity provider. In the next scenario, it is considered that SP1 and SP2 are both identity providers and consumers. SP3 is still only a consumer. This setup is presented in the following picture:

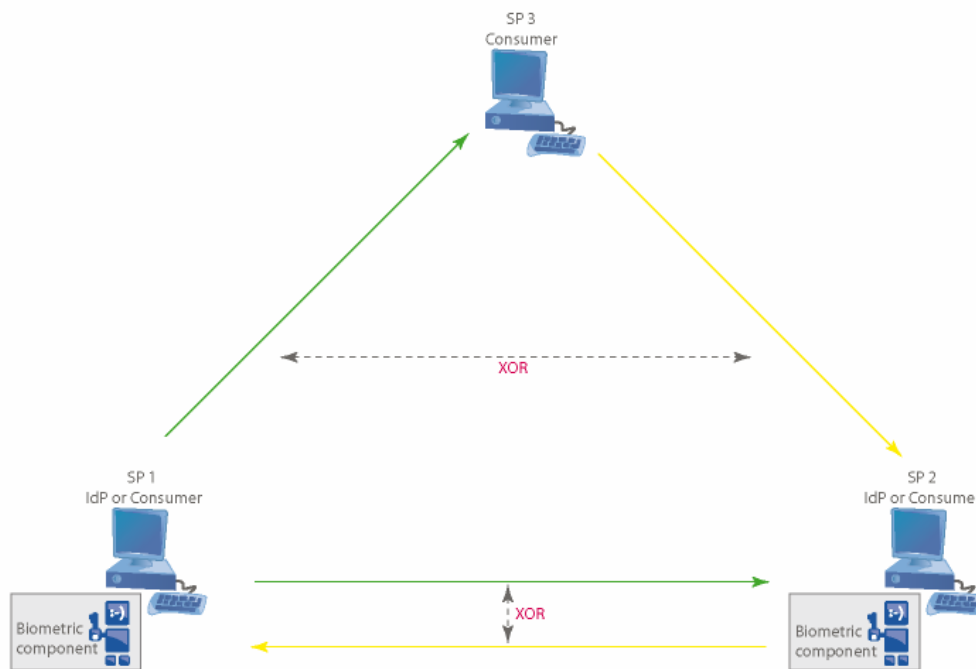


Fig. 10-11 The second configuration

In this case, the user can decide himself whether he will use SP1, SP2 or SP3 as identity providers. Should he decide, for example, to use SP2, he can also decide on whether SP2 can be used as identity provider for SP1. This leads to the same scenario that was described before, with SP2 as the identity provider.

The difference is that SP1 can also be an identity provider for SP2. In this case, we would reach a high localization with two identity providers in the same circle of trust. The logic flow is the same as before for enrolment and authentication, as SP1 and SP2 have the role of consumers or identity providers, depending on the order of enrolment and authentication.

The attribute update can be made in the following order:

SP3 (Consumer)  $\leftarrow$  SP2 (IdP, Consumer)  $\leftarrow$  SP1 (IdP)

In this case, when attributes are changed at SP2, it must update SP3 using the update\_URL function (unsolicited authentication response) and sending a store request to SP1. A store request is an authentication request with an additional parameter.

If the profiles must be immediately synchronised, this can be made when the service provider is also an identity provider.

As the SPs have both the functionality of consumer and identity provider, it is necessary to update their database model, in order to avoid redundancy problems. For this, we consider the original structure of the database model for an identity provider:

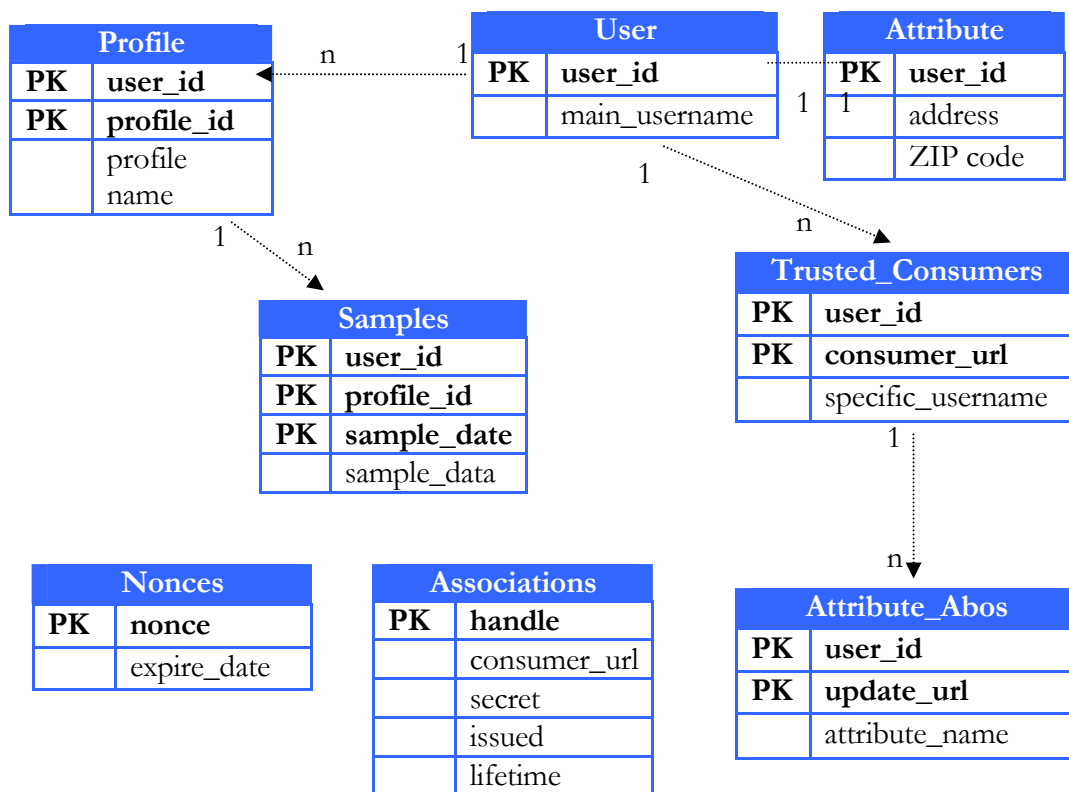


Fig. 10-12 Original database structure of an identity provider

A user stored in the <user> table can have more <profiles> with several <samples>. Each user account has a list of <trusted\_consumers>, a set of identity providers trusted by the consumer. Each consumer has an <attribute\_abos> table, which is a list of attributes that must be updated. In this example, the attributes are represented by biometric samples. The “update\_url” field from the same table informs the identity provider where it should send the changed attributes.

The <associations> table contains shared secrets with other consumers, which are used to sign the assertions.

Finally, the table <nonces> also carries an important role, as it is meant to protect the IdP from replay attacks: the identity provider generates a one-time ticket with each assertion. Every ticket has a unique time stamp. The consumer can check whether it has received such a ticket in the past. In this case, the consumer would recognize that the assertion is in fact a replay attack.

The database structure of the consumer has a structure similar to the IdP:

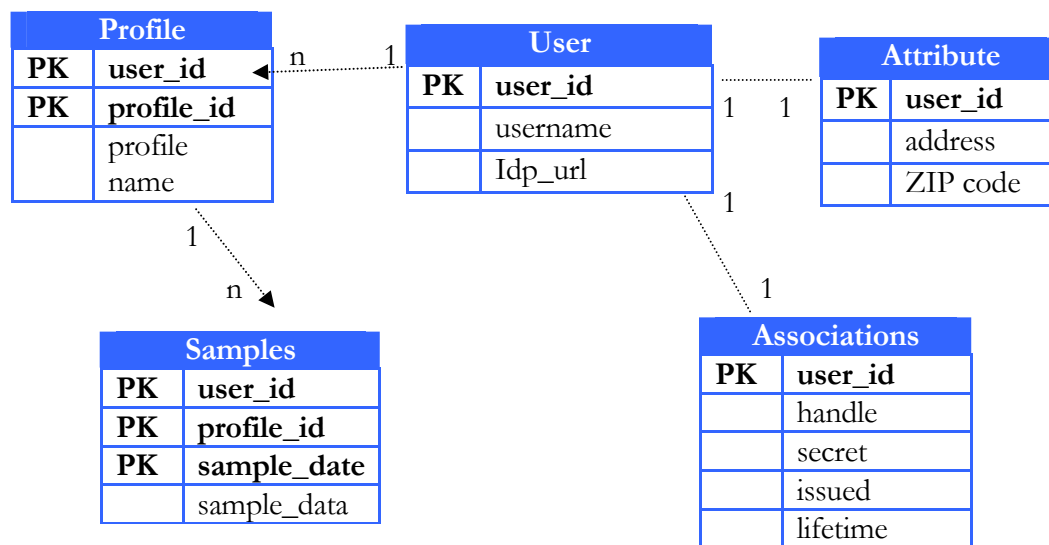


Fig. 10-13 Original database structure of a consumer

The differences between the two schemas are that the consumer does not require tables for <trusted\_consumers> and <attribute\_abos>.

A decisive argument is also the fact that it is possible for a consumer to have a shared secret for a certain user, therefore the consumer can send a handle in the authentication request; by means of this handle the IdP knows with what shared secret it has to sign the response. Due to this, the “smart mode” is possible in OpenID only when the communication has been started by the customer.

In order to realize the second configuration, it is necessary to overlap the two databases of the consumer and of the identity provider.

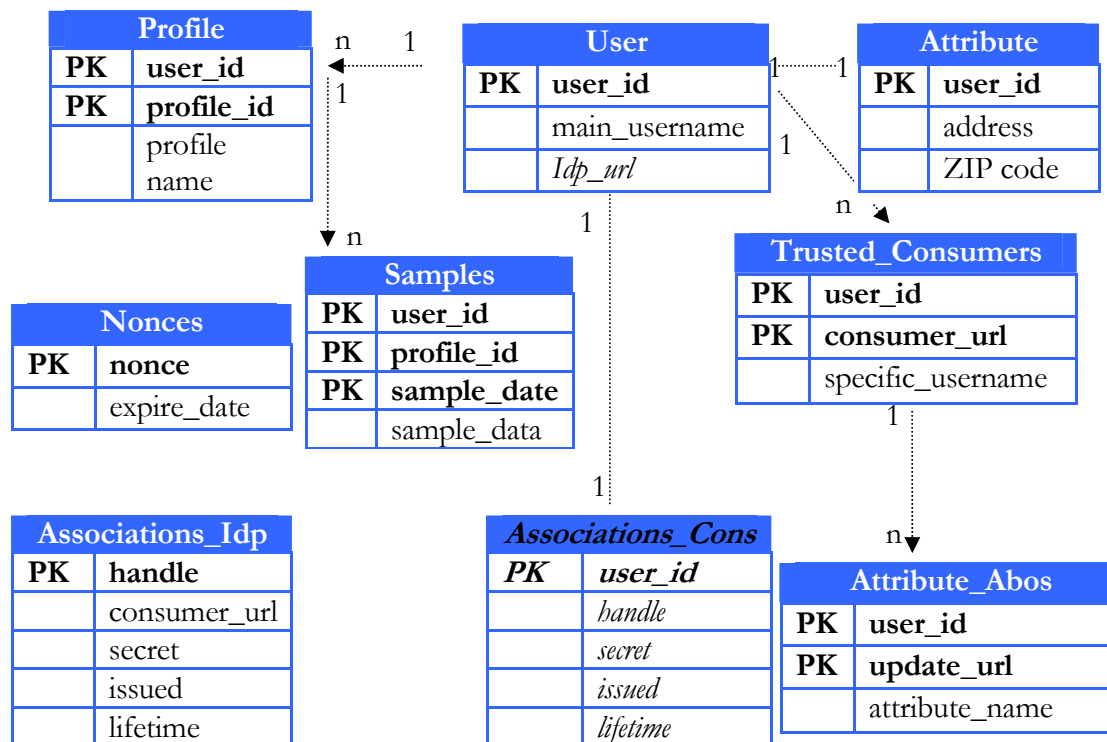


Fig. 10-14 Combined database model

Note: The changes marked *Italic* have to be made in the database structure of the identity provider in order to achieve this configuration. These changes are made only in the <association\_cons> table which is connected to the <user> table. If the application acts as a consumer, it can select a shared secret from the <associations\_cons> that can be used for communication with an identity provider.

### 10.5.3 3<sup>rd</sup> configuration: a user has several IdPs that have also consumer functionality

The following scenario is an extension of the previous use case, where we also used applications that have both consumer and identity provider functionality. The assumption made there was that in their relation, the applications always have one fixed role defined by the process of enrolment. The following configuration assumes that IdP1 and IdP2 communicate in a more flexible way and exchange their roles as identity provider or consumer.

This scenario has a very practical purpose, which is important for biometric authentication: the profiles can be synchronised in real time, as a consumer can take the place of the identity provider in order to inform the other parties about changes by means of an unsolicited response.

This setup can be found in the following picture:

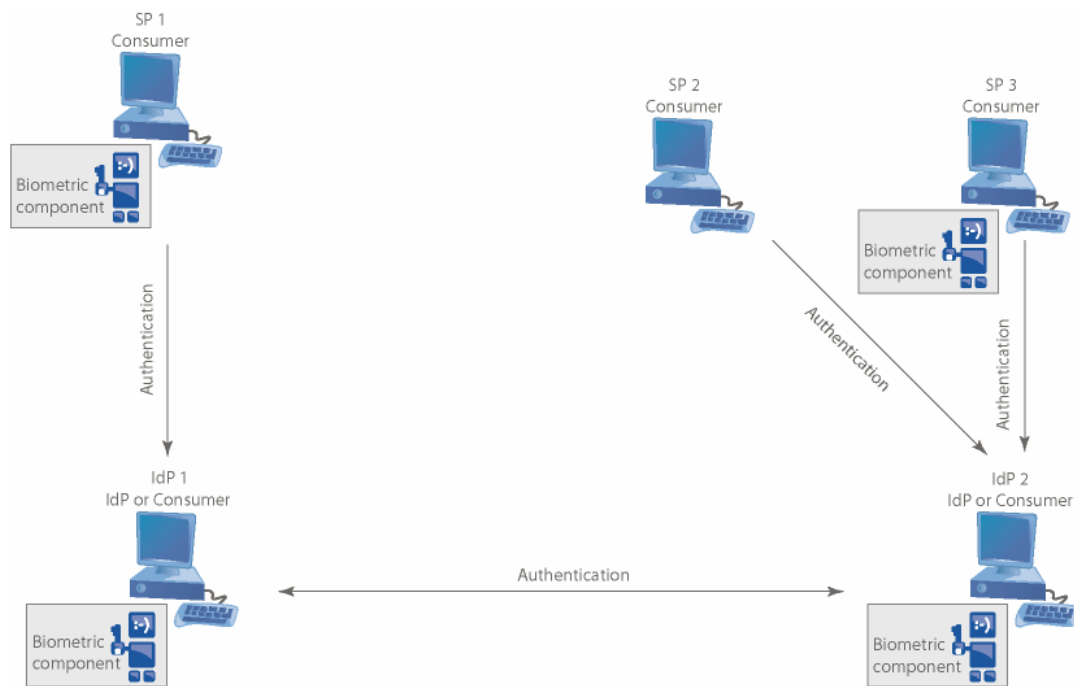


Fig. 10-15 The third configuration

### 10.5.3.1 Enrolment workflow

As IdP1 and IdP2 are also consumers, it does not matter where the user enrolls first. The logic flow is the same in both cases: first the user enrolls at one IdP and then registers at the second one, which has received, in his function as consumer, the biometric profile from the first IdP.

Following the logic flow presented before, one of the parties takes over the role of consumer and the other one acts as IdP. For the consumer, this means that he has to include the other party in the *<trusted\_consumers>* and *<attribute\_abos>* tables in order to achieve the functionality of an IdP. At its turn, the IdP must complete the *<association\_cons>* table and to know the URL identifier of the user that was given by the other IdP.

### 10.5.3.2 Authentication workflow

The login procedure and the respective attribute synchronisation do not fundamentally differ from the other scenarios presented so far. A decisive advantage of this scenario lies in the fact that the synchronisation can be done very quickly, even in real-time: if the user logs in to IdP2, IdP2 can assume the role of identity provider for IdP1 and send an unsolicited response, and vice versa. In

this case, the IdP1 as consumer does not have to wait for the user to log in to the identity provider IdP2 in order to send a fetch request.

For this configuration, the same database scheme as in the previous use case is applicable.

#### 10.5.4 4<sup>th</sup> configuration: a user can have more IdPs for a consumer

In the previously described configurations, we studied different use cases where a customer implemented a local authentication as a back-up solution for the biometric authentication provided by the IdP. For a light-weight consumer that does not use local authentication, such a case does not occur. In this case, a user registering to a pure consumer like SP3 is fully dependent on the functionality of its identity provider.

The following graphic shows a configuration where SP3 does not have its own user management, but allows the use of more identity providers. In this case, the user can take SP2 as identity provider in case SP1 is offline.

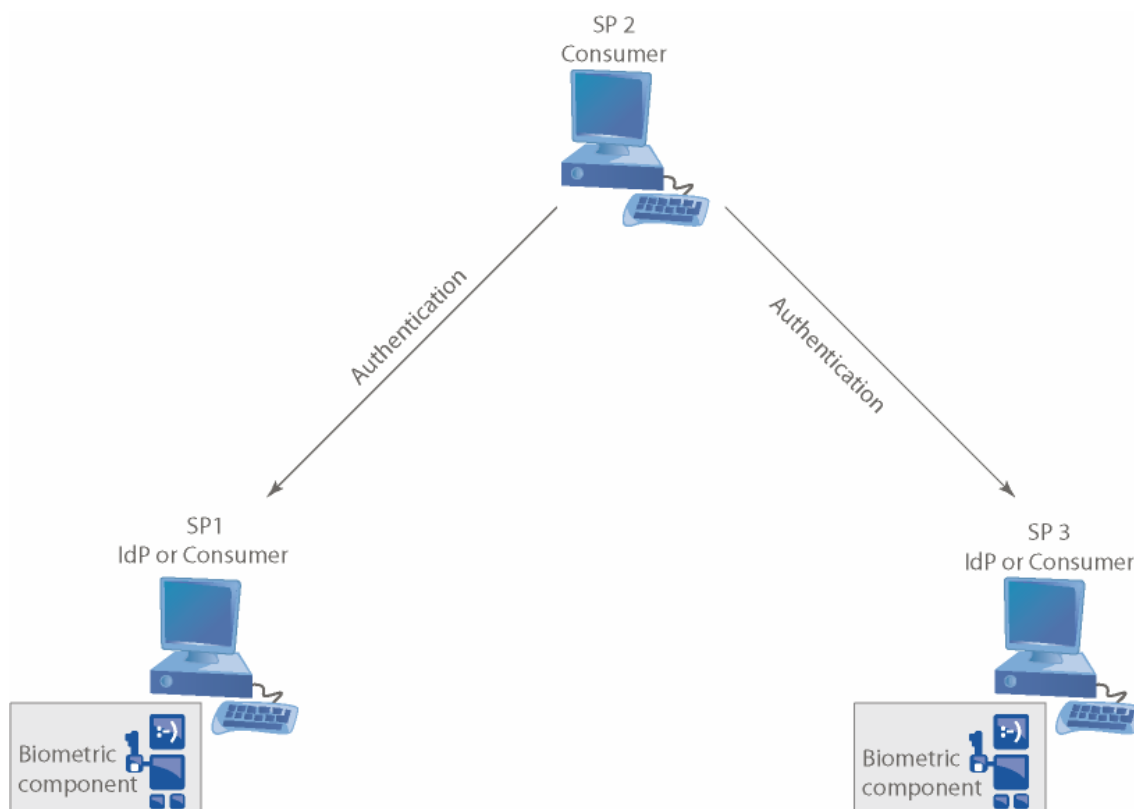
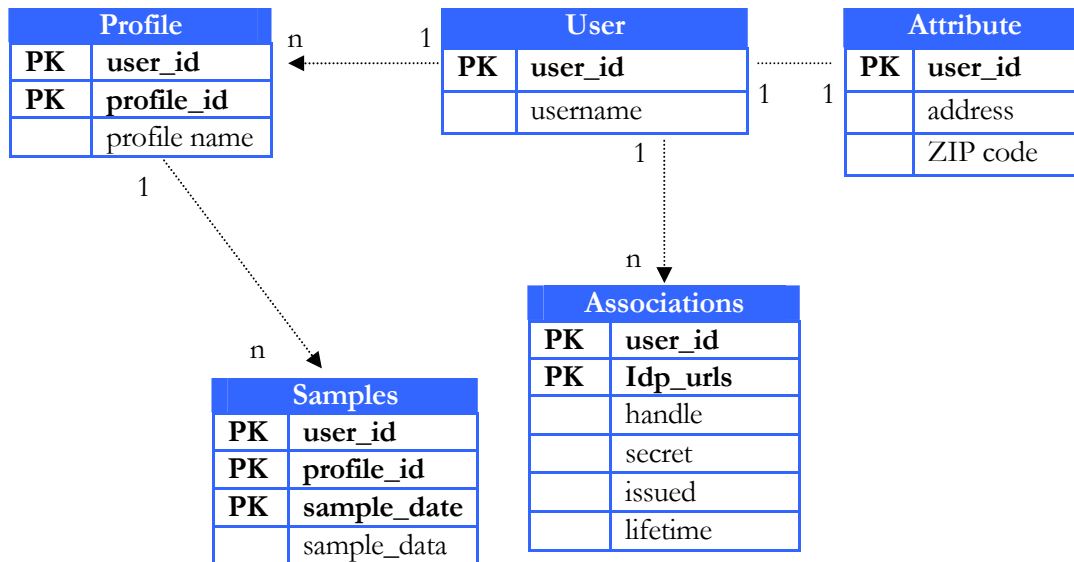


Fig. 10-16 The fourth configuration

For this, the database configuration has to be adjusted in order to allow several associations for a single user. This *1-n* relation between the <user> and the <associations> table is shown in the following image:



Furthermore, it is mandatory that the user does not authenticate at the consumer with his OpenID Identifier, but only via his username. This username is associated to several IdP URLs as shown by the associations between the “Idp\_url” fields in the <associations> table.

#### 10.5.5 5<sup>th</sup> configuration: an application supports all possible configurations at the same time

This configuration gives maximum flexibility, as it unites all the previously described scenarios. For example, the application SP1 can provide a good fall-back solution by allowing the use of more identity providers and by installing a local authentication. For this, it communicates with SP2 either as identity provider or as consumer. If SP2 supports configurations 3 and 4, the biometric attributes can be synchronised in real-time using the functionality of the identity provider in order to submit the new biometric samples by means of an unsolicited response.

This configuration is shown in the following graphic:

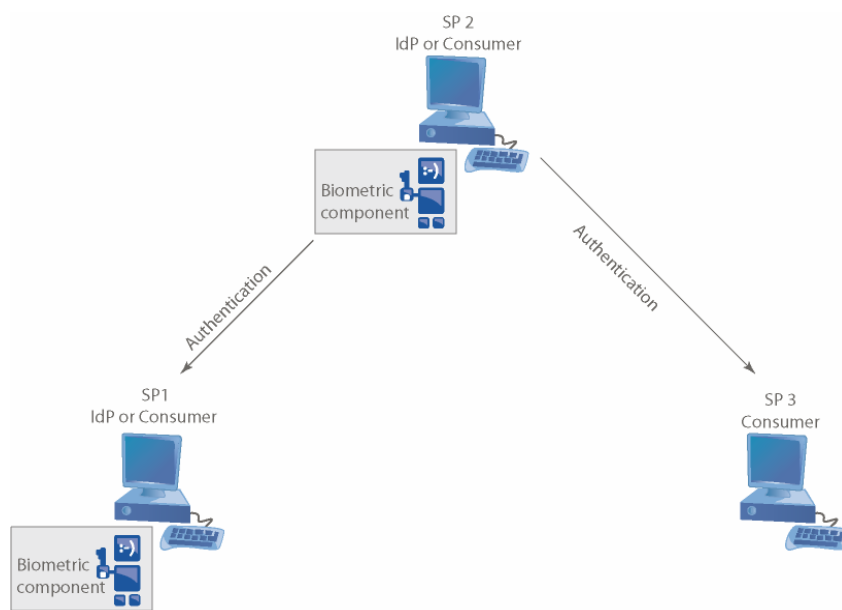


Fig. 10-17 The fifth configuration

The database structure required by this configuration unites the two previous database schemes:

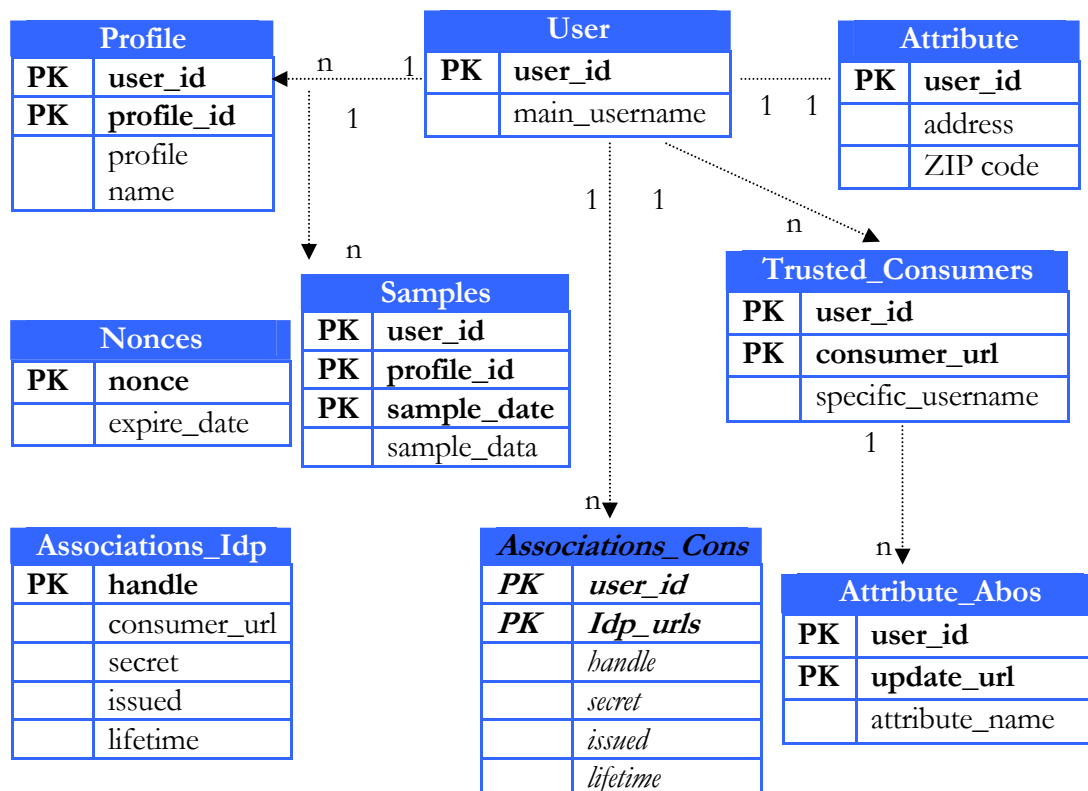


Fig. 10-18 Final database model

This database architecture does not require a large number of changes in the original and is fully compatible with the roles of both consumer and identity provider.

## **10.6 Conclusion**

The purpose of this chapter was to find different ways of sending biometric data in a Circle of Trust. For this, a synchronisation of the biometric data at database level was considered amongst other solutions. From a strict performance point of view, it is an interesting possibility but it also requires a stronger trust relationship between the parties. The user, in this case, has no control over the applications managing his attributes.

Another possibility presented is the data exchange at the level of the authentication and authorization infrastructure, which assumes a loose coupling between the parties and also grants the user control of his data.

Using OpenID as AAI in a circle of trust can be achieved with small changes in the framework and without losing the protocol compatibility. This configuration is also desired by other companies (Sun 2007); it is probable that OpenID development will allow this possibility in the future.

## *Chapter 11*

### **11 BIOMETRIC AAIS WITH REMOTE AUTHENTICATION**

---

While synchronising biometric data can be made both at database and AAI level, it requires a high degree of trust between the identity providers involved. This is usually not a problem when the IdPs belong to the same company, but it is an important obstacle for federations made between different providers. Another possibility that can be used in this situation is remote authentication, which assumes that the user stores his data at only one identity provider from the circle of trust. This method is researched by means of a biometric AAI prototype.

---

#### **11.1 Introduction**

As shown in the previous chapters, creating biometric AAIs and binding them in circle of trust structures leads to more problems, which were discussed before in this work. One solution based on the synchronisation of biometric data was elaborated. As synchronising biometric data can be made only with high constraints, another possibility for solving these problems has to be researched.

For this, we assume the following restrictions for the model:

A. A consumer works together only with one identity provider and has a trust relation only to this IdP.

If a user wants to access the services of a certain consumer, the user must be registered at the identity provider that works together with that consumer. For this, he stores his biometric identity at that provider by means of an enrolment process.

B. The identity providers, although they work together, do not have a trust relationship concerning the biometric data.

The different identity providers from a circle of trust work together, nevertheless they do not allow each other to access the biometric data of the users that are registered at their own service. Still, it has to be possible for a user to access the service of any consumer from the circle of trust, independent of the identity provider where that user is enrolled. This scenario is not possible due to the previous restriction.

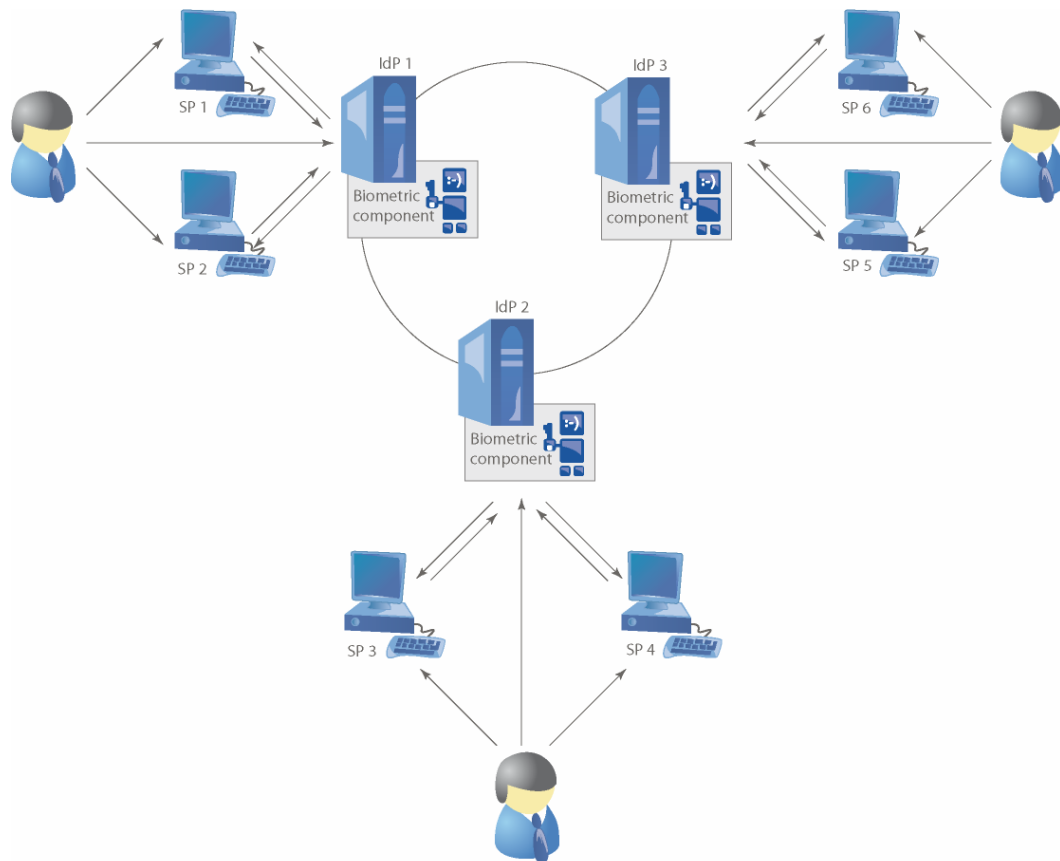


Fig. 11-1 Circle of trust with biometric AAIs

Let us consider the previous illustration as example: the user on the left is only registered at the IdP1 that works together with only two consumers SP1 and SP2. If he would want to use the services of the other consumers from the circle of trust, he would have to register at all other identity providers. This is possible, but as all identity providers use biometrics as authentication mechanism, following problems will occur:

- Aging: the biometric features of the user are changing, but these changes are not registered by the identity providers where the user does not login often;

- Replay attacks: should any biometric sample of the user come in the hands of a hacker during the authentication process at one identity provider, it is possible that the hacker authenticates with a copy of the sample to another identity provider from the circle;
- Quality of biometric feature: different identity providers may use different technologies and configurations for the biometric profile of the user, which would lead in the end to biometric templates of different quality.

C. These considerations lead to a final restriction, which states that as the user has only one biometric identity in his real life, he must have only one biometric identity in the circle of trust. No additional enrolment should be necessary.

## **11.2 Possible solutions**

This dilemma can be solved by means of a circle of trust, which is a union of several institutions that work together by means of defined contracts. By this it is assured that any user can access the services of all consumers, independent of the fact that the user is not registered at that particular identity provider the consumer works with.

A possible practical solution to this problem is given by OpenID itself. Although OpenID is designed to be used as a central Single Sign On server, it can still be integrated in a circle of trust infrastructure in respect to protocol conformity. This has been also done by SUN Microsystems, where OpenID is connected since 2007 (Sun 2007) in a circle of trust based on Liberty Alliance.

In order to extend the functionality of OpenID to support a circle of trust, we need to adapt two components of the protocol. One of them is the discovery process, which is a technique used to find the right identity provider that will authenticate the user. The other component that needs to be improved is the assertion process, which is the technique used to prove the authenticity of identity and of attribute information of the user between IdPs and consumers in the circle of trust. The changes in the discovery process are necessary due to the contract relationships that the different identity providers have in order to access the circle, while the assertion process must be adjusted in order to share the authentication data in the circle of trust.

### 11.2.1 Changes in the discovery process

For achieving the changes in the discovery process, four basic methods can be used:

- The consumer can choose the IdPs that make the authentication;
- The user can choose his own IdP and inform the consumer;
- The consumers must refer to a central instance in order to find the IdP of the user;
- An extra instance added to each IdP (CoT-Logic) will choose the right IdP for the user.

### 11.2.2 Changes in the assertion process

In case of changes in the assertion process, there are also four ways in which the assertion process can be modified in order to achieve a circle of trust structure:

- A central database in the circle of trust is used;
- Data mirroring between all the IdPs from the circle;
- Dynamical exchange of authentication data;
- Forward of authentication data to the right IdP.

### 11.2.3 Choosing the right solution

Combining each way of changing the discovery process with each possibility of adapting the assertion process leads us to 16 possible solutions for the realization of a circle of trust. Out of these, three exclude each other and will not be considered. For the rest, a set of criteria must be determined in order to choose the most efficient solution. These criteria must also have importance scores, which will be marked from *low importance* until *maximum importance*.

A. Data protection:

The circle of trust will store, additionally to the user data, the biometric features of every user. As this data cannot be replaced if it is lost, this factor is of maximum importance. In this case, we can distinguish between the following features:

- Data economy: the parties should exchange as little as possible data (especially biometric data);
- Data processing: if possible, this should be also reduced to a minimum;
- Control of the user over his data: as in OpenID the user has full control over his data, this should be preserved in the solution.

#### B. Conformity to OpenID protocol:

This criterion is also of very high importance. The conformity should be kept, if possible, to both 1.1 and 2.0 variants of OpenID.

#### C. Required trust in the parties that make the transaction:

Biometric data has higher trust requirements than any other user data. This trust implies the relationship of the user and the consumer and the choice of the right IdP to store user data.

#### D. Performance:

The system has to transport a small volume of data and has to be available all the time. If possible, it should not have a single point of failure. The system must be easily extendable with new IdPs and consumers.

#### E. Software changes:

This is a criterion of least importance in relationship to the others mentioned above and involves the changes that have to be made at the level of consumer, home and remote IdPs, as at the level of the user.

The criteria and their importance levels are presented in the following table:

Criterion	Importance
Data protection	Maximum
Protocol conformity	Very high
Required trust	High
Performance	Middle
Software changes	Low

Table 11-1 Criteria for designing a circle of trust with OpenID

Based on these criteria, the possible solutions can be ranked and sorted as follows:

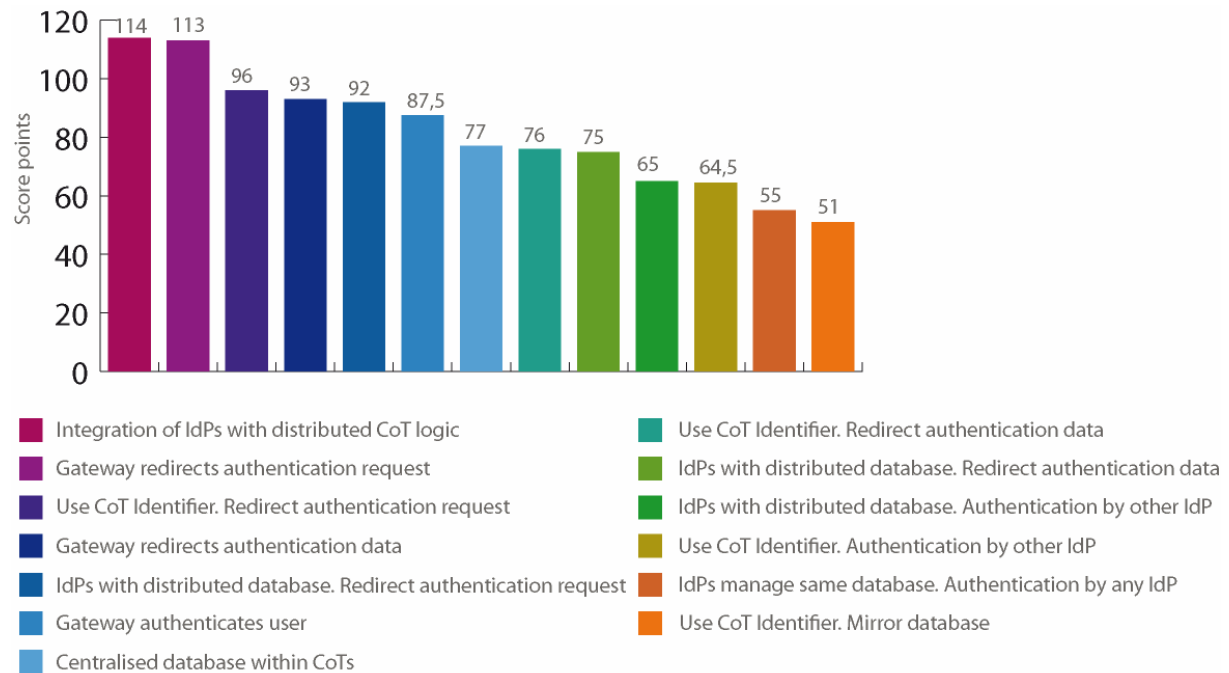


Fig. 11-2 Ranking process of possible solutions

The solution that meets most criteria presented before is the integration of IdPs by means of an instance that works together with each IdP and can redirect authentication requests. We will call this instance “CoT-Logic”.

### 11.3 The CoT-Logic

The CoT-Logic is a software module that has the task of constructing and administering a biometric circle of trust. Its task, technical features and use will be presented in this chapter.

For this, we use one of the assumptions made before, which states that consumers have preferences about the IdPs that make the authentication of users. These preferences are based on special trust in the authentication system of a certain IdP or by means of well-determined contracts. The consumers require that all users that want to access their service should be authenticated by their preferred identity provider. In case this is not possible, for example when a user is not registered at their preferred IdP, he will be rejected, although the IdP that is responsible for the user is trusted and correctly authenticated. The consequence for the user is the fact that he must register at the preferred IdP of that consumer in order to use its services, a process which is not only much more difficult due to the additional enrolment phases generated by the use of biometrics, but that can also present several problems mentioned in chapter 5.

In order to remove this effect, the IdPs can join a circle of trust, which allows to each IdP to confirm the authenticity of users to consumers participating in the circle of trust.

The technical basis for building and managing the circle of trust is the only task of the CoT-Logic, which does not specify nor implement any other protocols, such as a possible data exchange between IdPs or the authentication process itself. The CoT-Logic is meant to function as an add-on to each IdP and it will be installed on the web server of each IdP, which means that each IdP is managing his own CoT-Logic instance. The CoT-Logic instances have the task of registering IdPs in the circle of trust, if necessary to remove them from the circle or to inform each other about possible changes in the structure of the circle.

In order to exchange messages between each other, the CoT-Logic instances must use specified message formats with asymmetrical cryptography. For this, each CoT-Logic must have a public and a private key. The division of the public keys must be organized by the administrators of the servers.

In order to meet the assumption that consumers have preferred IdPs, each CoT-Logic instance is forced to inform the other similar instances from the circle about the consumers that prefer the home IdP where the CoT-Logic is running. For this, the CoT-Logic instances store lists of the consumers with whom the IdPs is working and must keep these lists synchronised.

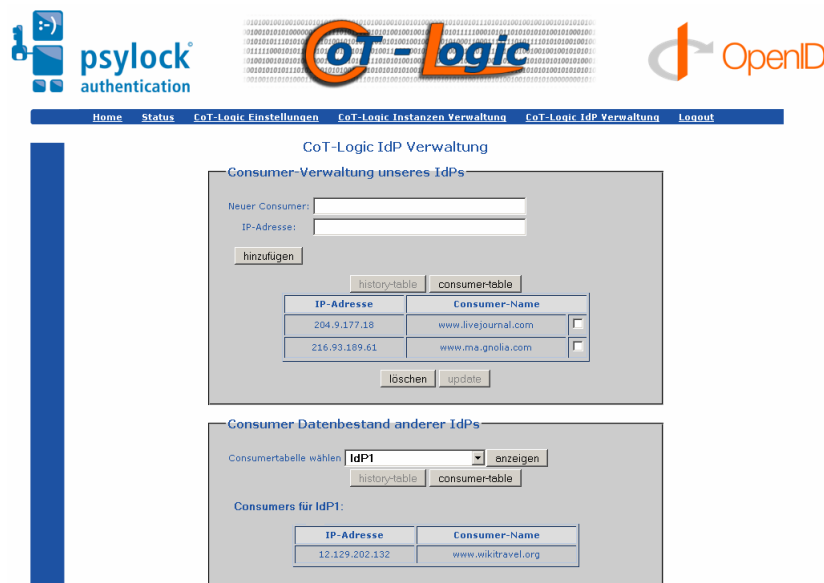


Fig. 11-3 The CoT-Logic

When a user logs in with OpenID to a consumer, he must give the address of his identity document. For reasons of simplicity, the CoT-Logic assumes that the identity document of the user is hosted at the IdP who issued it. Should the consumer request the identity document, it will be recognized by the CoT-Logic instance of the IdP by means of his IP address. The CoT-Logic will then dynamically create a modified version of the identity document especially for the consumer that requested it. In the new identity document, instead of the address of the users' IdP, the CoT-Logic will insert the address of the preferred IdP of the consumer.

The next example shows an identity document for the owner of the URL

`www.uni-regensburg.de/?user=matthias`

```
<html>
<head>
<link rel="openid.server"
href="http://www.exampleidp.de/authenticate/">
<link rel="openid.delegate"
href="www.uni-regensburg.de/?user=matthias">
</head>
<body>
</body>
</html>
```

The preferred IdP of the consumer will be dynamically inserted in the “openid.server” tag of the document.

The functionality of the CoT-Logic described before allows the discovery process of the IdP responsible for a certain consumer and it is transparent towards consumer, users and the IdPs that take part into it.

Based on this knowledge, we define the CoT-Logic as a software module that runs on a server and that has the task of recognizing the consumers that ask for the identity document of a user and dynamically generate for them their preferred IdPs instead of the standard IdP of the user, by means of modifying the identity document. The consumer will contact its preferred IdP and thus find an entry point in the circle of trust.

Should the user not be registered at the preferred IdP of the consumer, the authentication will have to be made in remote mode by the IdP of the user. This procedure will be explained later in this chapter.

### 11.3.1 Ways of using the CoT-Logic

When designing the CoT-Logic, we can consider two possible ways of installing and using it. For once, the CoT-Logic can run in a “standalone mode”, that is it can have two components, one that is responsible for the modification of the identity document and that be installed outside the circle of trust and another one that is managing the customer lists on the server. This method has the advantage that the user has the full control over his identity document, as specified in the OpenID protocol, but on the other side, the user must manage an extra component, thus making the authentication more complicated.

Another way is to use the CoT-Logic in a “server mode”, as an add-on module for OpenID, which means that the identity documents are stored on the same server as the IdP and the CoT-Logic. The user has less control over this document, but the login procedure remains unchanged.

In the following, examples for the use of these two CoT-Logic variants will be presented.

#### 11.3.1.1 CoT-Logic in standalone mode

In this case, the user manages his respective CoT-Logic instance on his web space. The logic flow of this variant is presented in the following graphic:

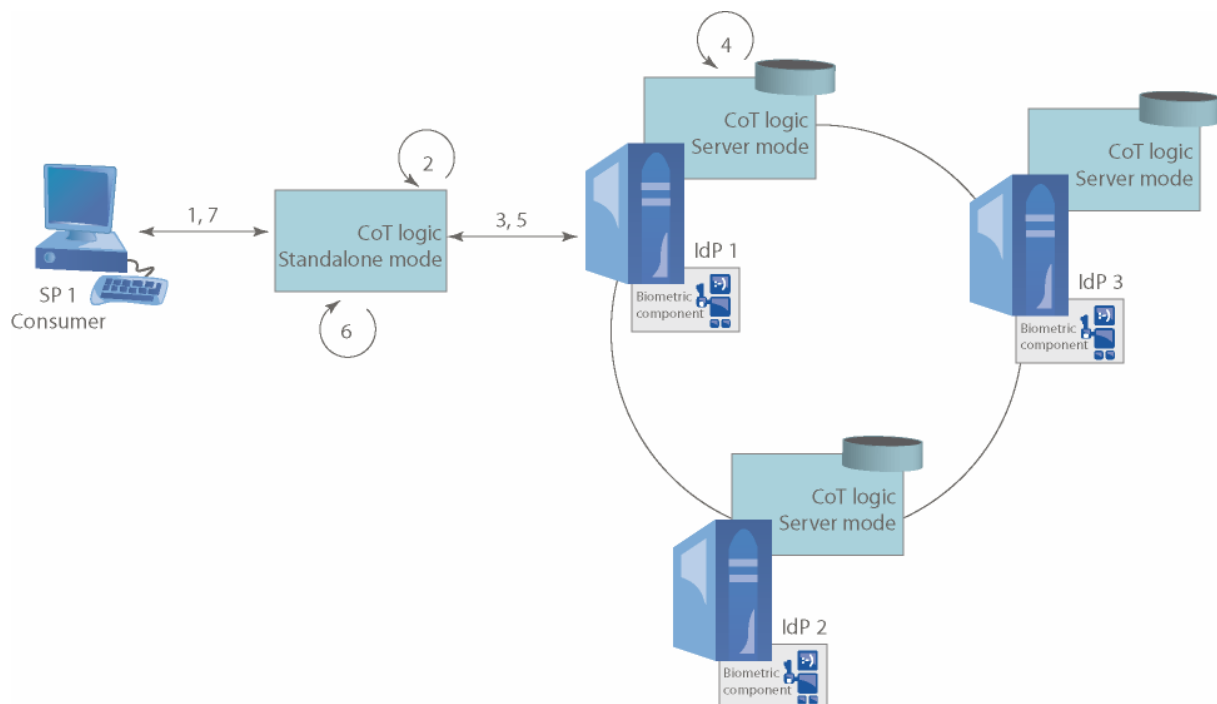


Fig. 11-4 Logic flow of the first CoT-Logic variant

1. The consumer calls the User Supplied Identifier (e.g. www.UserName.com)
2. On his web space runs a CoT-Logic instance in standalone mode. The CoT-Logic extracts the consumer that requested the document.
3. The CoT-Logic instance in standalone modus asks its server instance which IdP is responsible for the customer.
4. The CoT-Logic instance in server mode looks up the right IdP for the customer in the database.
5. The CoT-Logic in server mode sends the found IdPs (one or more of them) to the CoT-Logic instance in standalone mode which is running on the web space of the user.
6. The CoT-Logic in consumer mode chooses one of the IdPs and generates the identity document.
7. The identity document is delivered to the consumer.

#### 11.3.1.2 CoT-Logic in full server mode

In this case, the two components of the CoT-Logic are installed on the same server as the IdP. This component has also a database connection. The logic flow is presented in the following graphic:

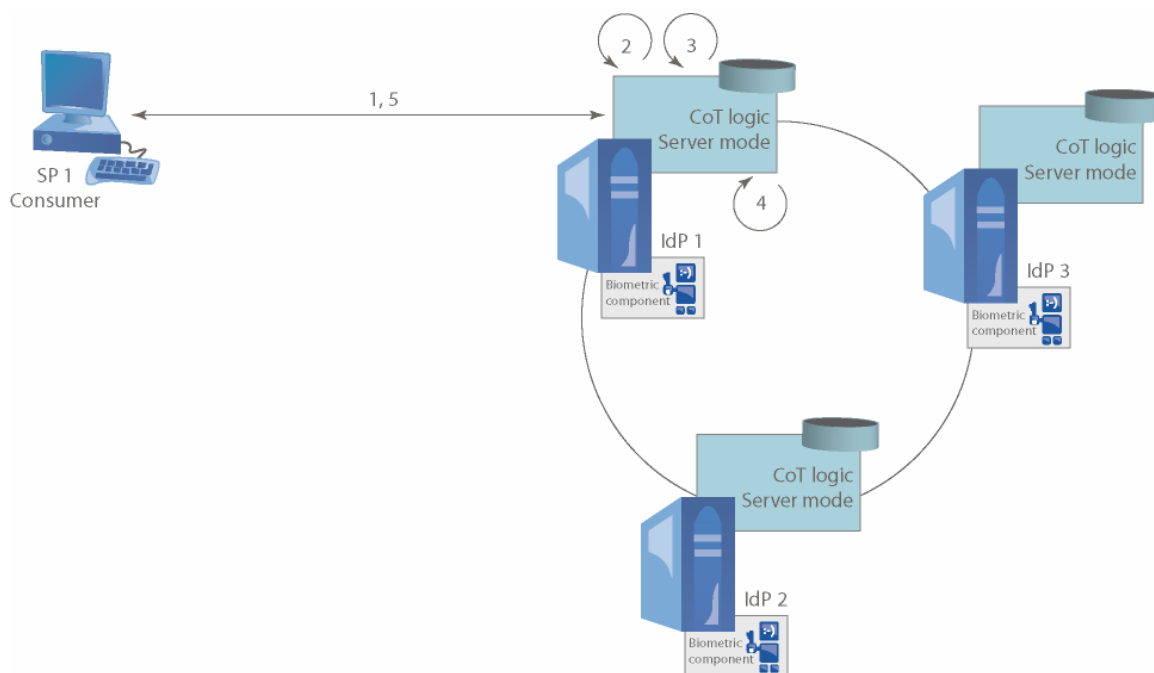


Fig. 11-5 Logic flow of the first CoT-Logic variant

1. The consumer calls the User Supplied Identifier (e.g. `www.idp1.de/?user=matthias`)
2. An instance of the CoT-Logic runs on the web space in server mode. This instance extracts the customer that requested the document.
3. The CoT-Logic looks up in the database the IdPs responsible for this customer.
4. CoT-Logic chooses one of the IdPs and generates the identity document.
5. The identity document is delivered to the customer.

From the technical point of view, both variants are possible but as the second variant (full server mode) is more comfortable for the user, this version of the CoT-Logic was implemented in the final prototype.

### 11.3.2 Division between the CoT-Logic and the IdP

Designing the CoT-Logic as an additional component attached to the IdP involves possible problems upon the update of the software used by the IdP. Therefore, it is important to make a strict division between the components that work together with the CoT-Logic and the ones that belong to the IdP.

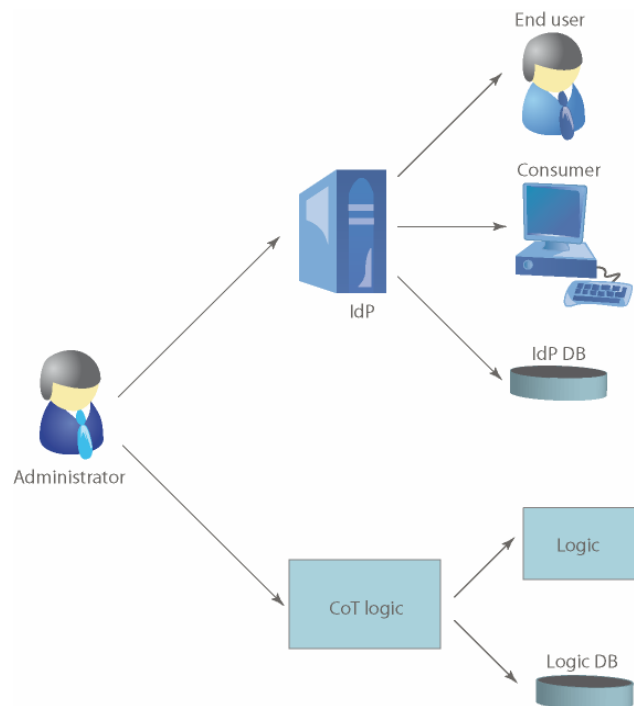


Fig. 11-6 Division between the CoT-Logic and the IdP functionality

As the previous figure suggests, the CoT-Logic instance and the IdP are fully separate systems and it is possible to manage a CoT-Logic instance without the existence of an IdP. The administration of both systems can be done by the same administrator, who can configure the list of other IdPs from the circle of trust. The CoT-Logic instance will contact these instances and ask for entries of customers that work together with them.

### 11.3.3 Data storing of the CoT – Logic instances

In the model presented here, a CoT-Logic instance is used for each IdP from the circle. This instance has its own database with different tables. The functionality of the main tables from this database is presented as follows.

*“Consumer”* table:

Each IdP must have a table where the names and IP addresses of the consumers that work with it are inserted (consumer table). The “c\_ip” column contains the consumer IP address and the “c\_name” column stores the consumer name in form of its domain address.

*“Integration”* table:

Also, each CoT-Logic instance has an integration table that stores the relationship between the consumer table of the Idp and his Endpoint URL.

*“History”* table:

The CoT-Logic instances have also tables where the changes in the consumer tables are saved (history tables). This table contains the columns “type”, “c\_ip”, “c\_name” and “timestamp”. In “type” we store the action (create, update or delete); in the “c\_ip” and “c\_name” we save the IP address and the domain name of the consumer, while “timestamp” stores the time of the actual change.

*“Fast index”* table:

The creation of a consumer table for each IdP allows a good overview of the system, makes the synchronisation process between the CoT-Logic instances easy and also simplifies the administration. The disadvantage is that when looking up the respective consumer all the tables have to be verified, which makes response times bigger. For this reason we use a fast index table

for storing the consumer IPs and the corresponding IdP endpoint URL. When the “consumer” table is changed, the modifications are also made for the “fast index” table.

“*Logic*” table:

Each CoT-Logic has a “logic” table, where data from other CoT-Logic instances is saved. This table has the following columns: “logic\_name”, “logic\_ip”, “logic\_pubkey” and “logic\_status”. The “logic\_name” and “logic\_ip” column contains the name of the current CoT-Logic instance and its IP address. The “logic\_pubkey” contains the public keys received when joining the circle. These keys are necessary in order to prove the signed messages sent by the other CoT-Logic instances. In the “logic\_status” we can have the following values:

- *activate* (for a CoT-Logic instance which was received but not yet confirmed by the administrator);
- *awaiting\_activation* (same case, but when the other instance does not respond);
- *ok* (after activation).

“*Log*” table:

Each CoT-Logic instance has a “log” table, where the system events of the program are saved. This is necessary for error recognition or for monitoring activities.

“*Check replay*” table:

Finally, each CoT-Logic instance has a “*check replay*” table. Its functionality will be explained in the next subchapter of this work - “Secure communication”. For the moment, it is important to mention that this replay operation must not be mistaken for the biometrical replay previously discussed in chapter 6.

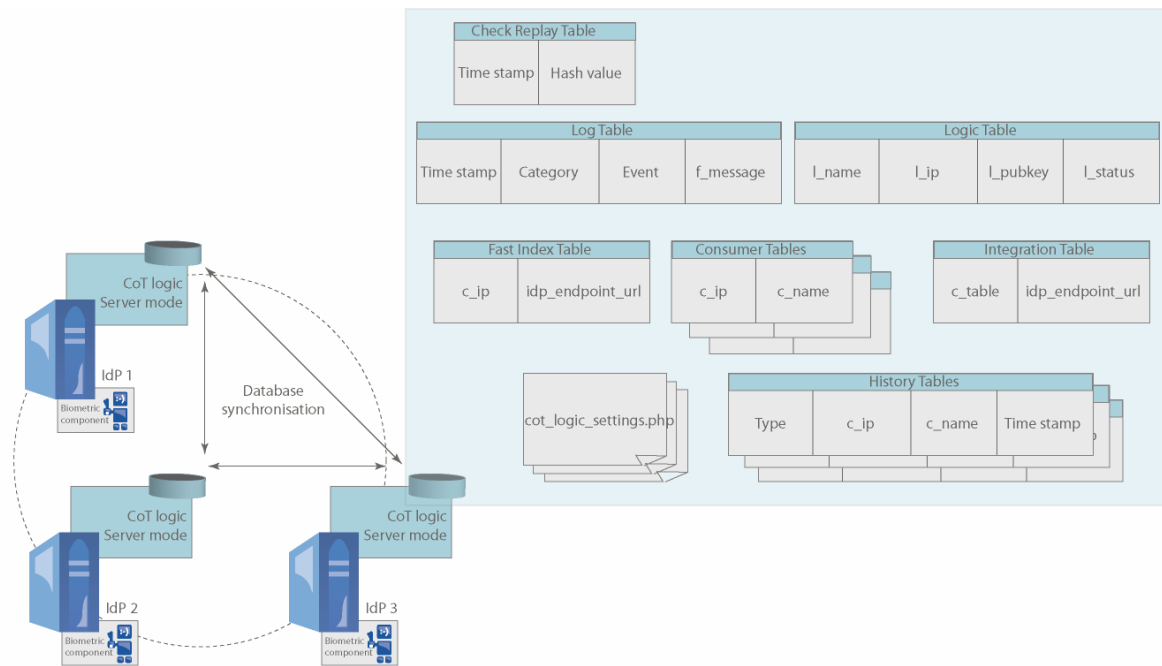


Fig. 11-7 Data storage of the CoT-Logic instance

The CoT-Logic also includes other, additional tables which provide basic configuration settings, like passwords or administrator e-mails.

### 11.3.4 Communication of CoT-Logic instances

#### 11.3.4.1 Secure communication

Each CoT-Logic instance has a public and a private key. The messages of the instances are being signed with the private key of the sender and checked for integrity and authenticity by the receiver with the public key of the sender. For this, the public keys of each instance are fixed by means of established contracts.

In order to prevent replay attacks by means of resending the same message from the other CoT-Logic instances, a timestamp is attached to each message before sending it. Each instance uses a “check replay” table which stores the hash values of the received messages and the according time stamp value. The data from this table is deleted after a predefined time frame.

Upon receiving a new message from another CoT-Logic instance, following steps have to be made:

1. Check the signature of the new message;
2. Check whether the timestamp is within the predefined time frame;

3. Check whether the time frame is still stored in the check replay table;
4. Store the message hash and the time stamp in the table.

If any of the previous steps does not return a positive answer, the message will be discarded.

#### **11.3.4.2 Consumer management**

When the “*consumer*” table of an IdP changes, the CoT-Logic submits these changes to the other instances, in order for them to actualize their “*consumer*” tables. When the data was actualized in the database of the CoT-Logic instance, a broadcast message is sent to the other instances. The broadcast message contains the new records from the “*history*” table, where the changes in the customer table were stored. All other CoT-Logic instances must adopt these changes in their own “*history*” tables and then to modify the “*consumer*” tables.

For the case that one instance was not available during the broadcast, the moment when it comes back online, it must ask all other CoT-Logic instances whether they have made changes in their “*history*” tables. These changes will be detected by means of the time stamp and then added to its “*history*” table.

#### **11.3.4.3 CoT-Logic instance management**

Adding a new CoT-Logic instance:

The process of adding a new CoT-Logic instance is done manually by the system administrator, in order to guarantee the human control over the involved CoT-Logic from the circle. The data inserted in this step must be stipulated by the contract made before between the companies.

If the administrator of the CoT-Logic instance named “A” is adding a new instance called “B”, the field “*l\_status*” will be set on “*awaiting\_activation*”.

The “A” instance then sends a signed message to the other instance “B”. This message contains the name of “A”, its IP address and its public key. The instance “B” can also add the first instance “A” in its “*logic*” table, this time with the status “*activate*” in the “*l\_status*” column.

The administrator of “B” will be informed of this operation by means of an e-mail. He must then check this step and confirm the new input. Upon confirming the correctness of the data, he will receive the status “*ok*”.

In the final step, the “B” instance will also send a message back to “A”, which will change the status of “B” to “ok”.

This procedure is shown in the next graphic:

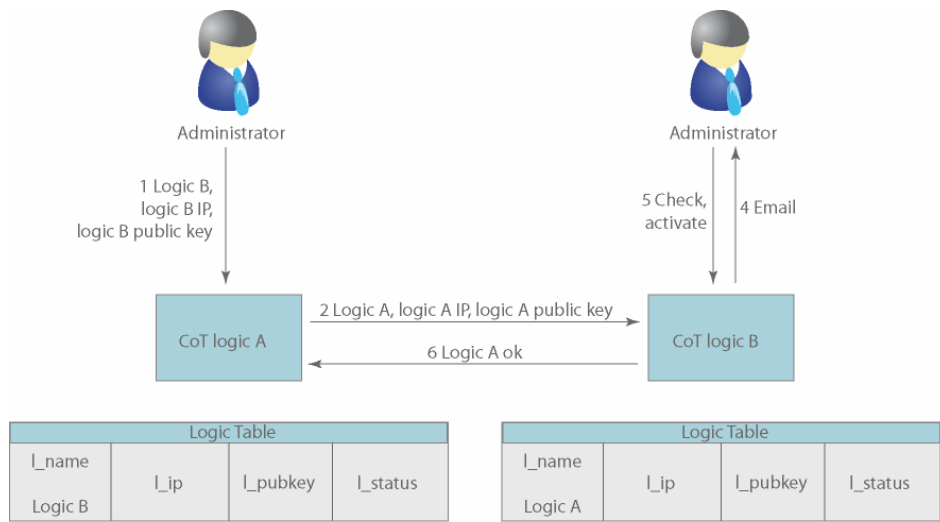


Fig. 11-8 Adding a new CoT-Logic instance to the circle

Deleting a CoT-Logic instance:

The administrator of a CoT instance A selects the instance that has to be deleted (for example the instance B). For this, the CoT-Logic A informs the other instance B about this process by sending a signed message to B. The message contains the name of the deleting instance (A), its IP address and its public key. Upon receiving this message, the administrator of B is also informed and he can start a similar process from the side of B.

11.4 Remote Authentication

While the CoT-Logic has the task of changing the discovery process in order to realize a biometric circle of trust, another process must be also adapted in order to meet the necessary requirements. The assertion process is modified by means of a “remote authentication”.

11.4.1 Definition

The “remote authentication” is both a method and a software system that have the purpose of adjusting the assertion process. This component is also installed as a module in OpenID. Starting

from the previously presented assumptions that a consumer trusts only a certain IdP and that IdPs do not synchronize biometric data, the task of remote authentication is to realize the authentication process when the current IdP does not have the user in its database. In this case, the current IdP uses the functionality of a consumer and sends the authentication request forward to the correct IdP, which then realizes the authentication assertion and sends it back to the current IdP, which finally confirms it to the consumer.

**psyopenid.uni-regensburg.de - IdP2**

[Home](#) [Log in](#) [Register](#)

You've tried to authenticate using a URL this server does not manage (http://psyopenid.de/?user=phfeustel). If you are using your own identity page, there may be a typo in the URL.

Username:

Password:

---

psyopenid.uni-regensburg.de - IdP2 | Contact [admin@example.com](mailto:admin@example.com)

Fig. 11-9 Problems without remote authentication

In order to realize the remote authentication, one possibility is to use standard communication protocols like SAML. Another way is to use a communication similar to the one used by OpenID itself. In this case, the current IdP (which does not possess the biometric data of the user) will take the functionality of a regular OpenID consumer in order to submit the authentication request to another IdP from the circle, where the user is already enrolled. We call this method “consumer mode”.

The design of this process can be divided into three main components:

- Remote authentication: This system is responsible for the management and control of the entire logic flow, for the communication with other components and for the embedding into the biometric OpenID server.
- Consumer mode: The enclosure of the whole functionality of an OpenID customer and the management of the database required for its proper functioning.
- Mapper: This component has the task of processing of authentication requests and answers at technical level and also the management of OpenID extensions.

The design and realization of these components will be presented in this chapter.

## **11.4.2 Functionality of remote authentication**

### **11.4.2.1 Integration**

The remote authentication process is triggered when an authentication request is decoded by the server. Then, it must be checked whether the user exists in the database of the current IdP and whether it is necessary to start remote authentication. This check is made by verifying whether the claimed identifier that the user submitted belongs to any user that the current IdP manages. From a technical point of view, this procedure consists in running a database query which will return whether the claimed identifier is stored in the database. If this is the case, the logic will abort and the normal functionality of OpenID authentication will be resumed. If the claimed identifier cannot be found in the database, the server will switch to the consumer mode in order to pass the authentication request to another IdP.

### **11.4.2.2 Checking the foreign IdP**

For performance and data protection reasons, it is interesting to investigate whether it makes sense to switch to consumer mode without any check, whenever the IdP receives a claimed identifier that it does not manage. In these circumstances, it is possible for an attacker to introduce a claimed identifier that points to an untrusted IdP that resides outside the circle. If the current IdP does not verify to whom it sends forward the request, it is possible to accept untrusted assertions from IdPs outside of the circle of trust.

When the current IdP starts the remote authentication, it receives an endpoint URL of the real IdP by means of discovery process and claimed identifier. For this, it is important to check the endpoint URL to ensure that it belongs to a trusted server. This step can be taken over by the CoT-Logic. The consumer mode sends a validation request to the CoT-Logic by submitting the newly received endpoint URL. The CoT-Logic has evidences about all valid IdPs from the circle and can check whether the URL belongs to one of them. After this, the CoT-Logic sends back the answer in form of a signed XML document that contains a “verification decision” of the endpoint URL and the decision as Boolean value.

This solution is highly flexible, as the remote authentication process can trust that the foreign IdPs that will make the authentication are belonging to the circle. Additionally to that, there is no need for this component to store its own list of all IdPs.

### 11.4.2.3 Representation of assertion relationships

Another challenge to the remote authentication is to present in a clear and transparent way for the user which IdPs are responsible for the assertion of his identity and for the forwarding of his attributes to the consumer. According to the OpenID protocol (OpenID 2008), the IdP has to use two parameters, “openid.realm” and / or “openid.return\_to” in order to show to the user which IdP is forwarding his attributes. These parameters must be bound in the user interface when the user decides whether to trust the consumer and share his attributes.

In case of remote authentication, the authentication request is not made by the original consumer, but by the IdP with which the consumer is working. This IdP submits the request in consumer mode. The user is informed about this process and asked whether he agrees to trust the IdP of the consumer in order to redirect his request to the home IdP of the user.

In order to resolve this problem, we can use the two parameters mentioned before. The “openid.return\_to” parameter can be used to describe the address of the IdP of the consumer, while the “openid.realm” parameter will point to the consumer itself.

However, the specification of OpenId states that the “openid.return\_to” URL must match the “openid.realm” (OpenID Authentication 2.0 2008). For this, we choose the solution of adding another parameter called “original\_realm”, which is a copy of the “openid.realm” parameter. This gives us the possibility to use this parameter without leaving the OpenID protocol, but requires also that the IdPs should be slightly modified in order to parse this parameter.

### 11.4.3 Consumer mode

If the process of remote authentication detects that the proof of identity must be made by another IdP from the circle, it starts a routine called “consumer mode”. This allows the IdP to act as a consumer upon presenting the request to the home IdP of the user.

In order to realize this component, a fully implemented consumer model from the OpenID framework (OpenIDenabled 2008) is required. The need for high enclosure is due to the fact that the framework for this consumer must be exchanged as soon as a new version appears, thus achieving a high maintainability.

From the design criteria of OpenID which uses redirects over HTTP or simple HTML forms, we conclude that the communication between the parties (IdP in consumer mode and home IdP of the user) can be made only asynchronous and with long time delays in the range of minutes. Therefore

it is recommended not to use threads in order to store the original authentication request of the consumer until the response of the home IdP arrives. Instead, it is better to store the original authentication request of the consumer in a temporary table.

At the same time, we see that a clear assignment must be made between the original authentication request of the consumer and the authentication request of the home IdP of the user.

#### **11.4.3.1 Mapping the authentication request of the consumer to the authentication response of the home IdP**

The authentication request of the consumer must be stored in temporary tables, due to the fact that the framework issues authentication responses only on the basis of authentication requests and a modification of the structure of the framework is not desired. The second reason is that it is not possible to build dummy authentication requests from the data of the authentication response of the home IdP.

If the authentication response of the remote IdP issues an authentication response, this will be mapped to an authentication request stored before in the temporary database of the consumer mode.

A comparative analysis of the request and response parameters in OpenID (OpenID authentication 2.0 2008) shows that only the parameters “openid.ns” and the “openid.mode” are always sent. All other parameters are either optional or only used by one single party, without giving the possibility of identifying the correct response for a certain authentication request. A mapping of these two functions cannot be done only by means of these two parameters.

Therefore, another parameter is used, which is generated by OpenID using the “Auth\_OpenID\_mkNonce()” function. This parameter is attached to the query string parameter “openid.return\_to” and called “ra.nonce”. The “ra.nonce” parameter will be stored in the temporary table together with the serialized authentication request until the authentication response arrives.

There is no need to use an incremental number for identification as numbers have their data type as limit and it can happen that they have to be reset, which can lead to problems in the system.

As the „Auth\_OpenID\_mkNonce()“ function must generate unique keys for

$N$  consumers \*  $M$  claimed identifiers

the complexity of its calculation is increasing.

Let us take an example of a nonce value generated by this function:

```
2008-05-17T15:06:53ZTG3819
```

This is corresponding to the specified OpenID format for the variable „openid.response\_nonce“.

By this, a time stamp is specified, which is precise to a second and to which is attached a combination of “six printable non white-space characters”. From this we receive  $62^6 = 56.800.235.584$  possibilities per second for the random part. This value makes this function appropriate for being used as a random generator.

#### 11.4.4 Mapper

This component has the role of dividing, processing and extracting the parameters from the authentication request and response as well as the alignment of the requests and response from the different parties involved. The mapper is installed as component of the IdP and submits, for example, the positive authentication response to the consumer upon the receive of a positive response from the remote IdP of the user. In the same way, the mapper processes the authentication requests of the consumer and sends these to the remote IdP.

Additionally, the mapper can be improved with different OpenID extension. For this, the mapper must be capable to manage these extensions and to connect them together.

From a theoretical point of view, it can be assumed that different IdPs have different versions of OpenID and therefore different extensions. A mapping between the different protocol versions and their extensions would be also of great interest. However, for the prototype of the biometric AAI we assumed that the IdPs use the same OpenID version and that all of them have the same extensions installed.

One of the extensions that can be mapped is the OpenID Simple Registration Extension, which allows the authentication request to ask for certain attributes (name, email, and so on) from the IdP and to receive them in the authentication response. This can be used, for example, to process all the required attributes that the consumer is asking and to adopt them in the authentication request made by the IdP in consumer mode. Also, the attributes returned by the home IdP of the user can be mapped back to the authentication response sent to the consumer.

The specification of this extension (OpenID Simple Registration Extension 1.0 2008) states that the attributes can be “required” or “optional”. The following additional situations are also to be considered:

- The remote IdP is delivering additional attributes than the ones required;
- The remote IdP is not delivering some or all of the required attributes.

The Simple Registration Extension also specifies that: “the consumer must be prepared to handle a response which lacks fields marked as required or optional. The behaviour in the case of missing required fields or extra, unrequested fields is up to the consumer. The consumer should treat this situation the same as it would if the user entered the data manually.” (OpenID Simple Registration Extension 1.0 2008)

This leads to the fact that, even if the IdP works in consumer mode, it has to leave all the decisions at the level of the original consumer and therefore to be fully transparent.

#### 11.4.5 Prototype demo

Based on the information presented in this chapter, the general logic flow of a biometric circle of trust with CoT-Logic and remote authentication can be modelled on the following example:

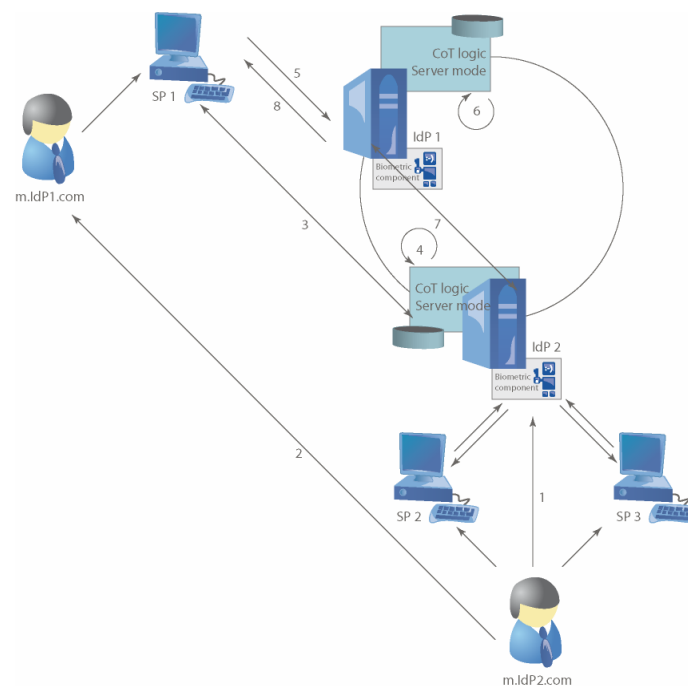


Fig. 11-10 Logic flow of the prototype

1) The user “M.” registers at the IdP2 and receives the OP-Local Identifier `www.m.idp2.com`. As this IdP uses biometric authentication, “M.” will have to enrol by typing a certain number of predefined sentences. The IdP2 works together with only two consumers, SP2 and SP3 and is member of a circle of trust together with IdP1.

2) “M.” wants to log in to `www.SP1` and uses a User Supplied Identifier like

`m.idp2.com`

where SP1 is a consumer from the same circle of trust which works together only with IdP1.

3) SP1 accesses the address `m.idp2.com`. The CoT-Logic receives this request.

4) The identity document must be dynamically generated. This is done by the CoT-Logic, which runs on the servers of the IdP2. The CoT-Logic must know with which IdP the SP1 cooperates, therefore it generates a new identity document with OP Local Identifier and with an OP Endpoint URL of IdP1.

```
<link rel="openid.server" href="http://www.idp1.com/authenticate">  
<link rel="openid.delegate" href="http://www.M.idp2.com">
```

5) SP1 parses the content of the delivered identity document and brings “M.” to IdP1 per HTTP redirect.

6) IdP1 parses the authentication request and realizes that it is not responsible for this OP Local Identifier.

7) IdP1 starts the remote authentication process. For this, it checks whether IdP2 is in the same circle of trust, then informs the user “M.” about this process. If “M.” agrees, he is forwarded to IdP2, where he can authenticate biometrically. Upon successful authentication, IdP2 submits its authentication assertion to the consumer mode of IdP1.

8) The authentication assertion is sent from IdP1 back to the consumer.

## **11.5 Advantages of using biometrics for the participating parties**

### **11.5.1 User**

The present standard for authentication in web applications is a combination of username and corresponding password (Computerwelt 2006). As a consequence, users can only access a web service if they register a new account and choose a password. The authentication through passwords is also used in other areas, like technical devices, applications, etc. Users are confronted with an increasing amount of passwords, which are becoming more and more difficult to handle. Through the application of OpenID within a circle of trust, users only need to register with one biometric identity provider where they generate their account with the respective data. In this way, users do not give away their biometric data in many different places on the internet but only to a single identity provider they confide in. This lowers the risk of data loss. With this one-time registration, users are given the possibility to potentially access all consumers connected to the circle and the effort of repeated registration can be dispensed.

The concept of the User-Centric-Identity is maintained in the implemented solution. It is only up to the users themselves to determine which of their profile data they want to release to different consumers. Users stay in control of their (biometric) data and can decide to whom and in which form they are passing it on.

The use of biometrics in combination with an IdP substitutes the authentication by password and therefore solves the problems related to this. The risk that an unknown person assumes the digital identity of a user is reduced, thus increasing the security of the user.

### **11.5.2 Identity provider**

The application of biometrics also holds advantages for the provider of an identity service. Identity providers ensure – usually in form of contracts – that they will authenticate users for the consumers in the circle of trust. By this, identity providers can achieve much more easily the security level demanded by the consumers. Possible consequences deriving from a failed authentication of an attacker in the name of a specific user can be minimized. A biometric identity provider differs from a simple OpenID provider who allows authentication only through passwords through the fact that it can provide a higher security during the authentication process; therefore it can have a distinct advantage over the other identity providers on the market.

Every biometric provider within the circle of trust sets up its own user database which is stored only on its servers. No data is passed on to other providers, e.g. through data mirroring.

Beside the possibility of offering the service for a fixed amount of money or for a given time span, other charging methods are possible, e.g. charging the user data check by volume or by the number of authentication assertions. This constitutes a very attractive payment method, as it brings low costs for consumers who have recently entered the circle and as it allows volume based reductions for big consumers.

### **11.5.3 Service provider (consumer)**

Every consumer has a defined biometric server that constitutes his IdP and introduction point in the circle of trust. For this, the consumer has an agreement with its IdP and profits from the innovative type of authentication used by the IdP, as users receive an easy and secure access to their services. When a consumer decides to participate in the circle of trust, it is able to immediately offer its services to all users who already hold biometric accounts. Through this, customers can concentrate on their core competences but at the same time offer services to more users in the circle.

As user management is outsourced to an IdP, consumers can lower costs and take advantage of the know-how and the security offered the IdP.

## **11.6 Conclusion**

This chapter presented a prototype implementation of a biometric AAI based on OpenID and Psylock. The IdPs from the created circle of trust are able to process authentication requests from consumers in conformance with the OpenID protocol (OpenID Developers Specifications 2008). Furthermore neither users nor consumers are supposed to be limited by the existence of the circle of trust. Neither should any additional activities for users and consumers be added to those that have already been specified by the OpenID protocol.

These advantages are achieved by the development of a software solution for the administration of the circle - the CoT-Logic. In case consumers accept only users of preferred IdPs, the CoT-Logic makes sure that consumers will be transferred to their respective IdPs in accordance with the OpenID protocol. If the users are not registered at the IdP where they were transferred by the consumer, the mechanism of remote authentication will make the IdP to take the role of a consumer and redirect the authentication request towards the IdP that stores the biometric data.

By means of remote authentication, every IdP can confirm the authenticity of every user of the IdPs towards every consumer from the circle. This is presenting advantages as it does not require data mirroring between IdPs and leaves the user in control over his data.

As the IdPs allow users to authenticate themselves through typing behaviour, the remote authentication also resolves the problems that come up when applying biometric authentication when users would have to register to several IdPs:

- Aging: As the user is always authenticated by only one IdP, he has only one biometric template which is aging more slowly.
- Replay-attacks: Replay attacks with a stolen sample of a biometrical characteristic are only possible at the home IdP of the user, not at other IdPs from the circle. As this IdP has all the typing samples of the user, replay-attacks can be recognized.
- Quality of the biometric feature: In case of only one identity provider, the user can have different biometric profiles and he can use different sensors upon the authentication to each customer. His profile always has the maximum quality that the biometric method can offer.

The prototype can be extended to function with different versions of OpenID and of its extensions. Another possible development is to transfer additional information about the quality of biometric authentication to the customers (for example the match score achieved) in order for them to decide to which parts of their web services they allow access for the user.

### **12 CONCLUSIONS AND FUTURE WORK**

---

This work has shown that biometrics can present a valid solution for a stronger authentication process within an AAI. A biometric AAI can be built only with strict consideration of the specific issues that come with the thematic; it allows only minor changes in the current logic flow. A prototype for a biometric AAI has been developed based on the typing behaviour biometric Psylock and the AAI OpenID.

---

#### **12.1 Conclusions**

This work presents the idea of improving the security of AAI systems by combining them with biometric authentication methods and provides both theoretical and practical solutions for implementing a biometric AAI.

Different trends in identity management have been investigated in order to realise this model. The result of this investigation was that the use of biometric authentication is seen as a possible tendency in the future, as it is fully compatible with the principles of the new “Identity 2.0” concept, which places the user and not the site in the centre of the system.

The choice of the biometric system to be used as a research model for the biometric AAI was easy as only the typing behaviour biometric implemented in the Psylock authentication recognition gives the possibility to use this biometrics in the web. Its full compatibility with different browsers and operating systems and the ease of use made typing cadence the only choice for a web-based biometric authentication at the moment.

The choice of the AAI system involved meticulous research on different methods and solutions available on the market. Based on a set of criteria (e.g. the availability of documentation and practical reference implementation as also the ease of installation and use even for small companies), OpenID was chosen as the platform for the biometric AAI. This decision proved to

be the right one, as in the two years of research, this system developed rapidly and it is now grown to be one of the most popular AAI systems in the web.

Two possible AAI architectures were investigated, the central Single Sign On and the Circle of Trust. A standard BIO-API compatible biometric system (BioAPI 2008) was applied to them. The conclusion was that several problems like replay attacks, quality and aging of biometric features present critical aspects in combination with AAI structures, especially the circle of trust.

It was therefore necessary to conduct a more in-depth research of these problems. Though all the previous research was made considering biometrics in general, these particular problems required that the investigation be done at the level of a specific biometric. The typing behaviour biometric method Psylock was selected for this purpose.

A first approach was the investigation of how replay attacks can be conducted for typing cadence biometrics; an appropriate algorithm has been designed in order to recognize and block such attacks. This algorithm was designed to work in the same way as a biometric method, which means that its quality can be measured by means of specific FAR-FRR-EER curves. The tests conducted in this work have shown how replay attacks can be recognized with a high probability.

Another set of tests was done in order to research the quality of a biometric feature. By means of different operating systems, browsers and keyboard types, the software and hardware limits of the typing behaviour have been tested. The result was several ideas for diminishing the negative influence of this factor.

The next consideration was the aging problem of biometric features. Experiments have shown which features of typing behaviour are the most subject to aging and what changes they undergo during this process. The results confirmed the supposition that typing behaviour ages very quickly. Therefore, the importance of this issue for biometric federation structures where users can access certain IdPs (with possibly outdated biometric profiles) only irregularly after long intervals of time is not to be neglected.

Although not foreseen originally, the use of an authentication based on two or more factors (password, biometrics, tokens) has shown the necessity of a fall-back mechanism. Such a mechanism had to be developed in order to have a functional system for the case that the user forgets his password, loses his token or the biometrics reject him. The conclusion was that such a mechanism must be based on more factors as well; else the fall-back variant would be less secure

than the authentication itself. The model presented in this work allows a fall-back for a two-factor authentication (password and biometrics) when one of these factors is compromised.

The knowledge acquired during the previous stages was used in order to develop possible solutions for biometric authentication. The conclusion drawn was the fact that when more IdPs synchronise the biometric data within a Circle of Trust, the probability of biometric problems decreases. The synchronisation mechanisms proposed in this work take into consideration the level of this operation: database or AAI; they also handle specific problems like the fact that the same user may have different accounts with different usernames within the Circle of Trust. This procedure was tested by means of a prototype and proven to be fully functional.

However, the synchronisation of biometric features involves a high level of trust among the involved parties, due to the fact that if a biometric feature is compromised, the user cannot replace it. Therefore, this work discusses a second model for a biometric AAI based on the premise that, as the user has only one biometric identity, he should also have only one identity within the Circle of Trust. This implies that only one identity provider is responsible for confirming a user's authentication. If other consumers that do not work with that particular IdP need to authenticate the user, this can be done through a process defined here as "remote authentication". This process is based on the IdP receiving the authentication request. This IdP can search for another IdP which stores the user's biometric information and forward the authentication request to this IdP. In the same way, the first IdP receives the authentication assertion made by the biometric IdP and sends it to the customer that has originally requested it. This second model of a biometric AAI was implemented by means of a fully functional prototype, which is modular and future compatible.

## **12.2 Future work**

This work presents a reference model for a biometric AAI and is based on minimal modifications at the level of the identity provider; wherever possible, it does not require changes from the various consumers. The use of biometrics as an authentication method gives a stronger binding of the username to the real person and therefore shows new opportunities for the cooperation of identity providers and consumers.

A possible future research topic is a possibility where the biometric IdPs do not answer with true or false authentication assertions, but send to the consumer the user's match score of the successful login attempt. The customer can then decide, based on internal policies, which parts of the online contents the user is granted access to, which was not possible before by using a password. This process is a "partial authentication" where the user or the identity provider can choose a more

comfortable authentication method, like for example a shorter text to type, which still provides good biometric recognition but allows the user the access to non-critical services of the AAI. In the same way, a stronger authentication can be required (for example by means of a longer text with superior quality features like less typing mistakes) when the user needs to access high security systems. In this way, a distinction can be made by the importance of different services offered by consumers. In this way, biometrics provides the solution to the problem of multiple access levels in a Single Sign On system, which cannot be achieved with the current systems. These new business cases opened by biometric AAIs have to be analysed in future work.

Then, using biometric AAIs involves the change of trust relationships between user, consumer and biometric identity provider; therefore it is important for the identity provider to accept biometric authentication requests only from trusted consumers. The changes in trust relationships and in the quantity of trusted information for each party are also a subject for further investigations.

On the side of biometric systems, a better replay attack protection can be achieved with a clear distinction between a real person that inputs biometric data in the sensor and a machine that does the same thing in an automatic way. An important topic here is live detection, for which at the moment there is not enough information.

Another important feature for biometric systems is the generation of a strong cryptographic key based on a biometric sample, which is currently not possible due to the fact that biometric samples change continuously. In connection to this, the aging process of biometric features should be investigated over a longer period of time (several years) in order to gain additional information about the changes that occur and possibly to develop mechanisms against this process.

A final interesting aspect that has to be further researched is the storage of biometric components at the user, for example in form of a software or hardware token. This structure would decrease the trust level the user has to overcome with consumers and identity providers, but it would raise completely new questions in the relationship of the parties involved.

## BIBLIOGRAPHY

Abels H., Identität, Wiesbaden VS Verlag, 2006

Achatz, M., Entwicklung eines Analysetools und Durchführung von Auswertungen für das biometrische Verfahren Psylock, Universität Regensburg, 2006

Amber, M., Fischer, S., Rößler, J., Biometrische Verfahren – Studie zum State of the Art, [www.wi3.uni-erlangen.de/fileadmin/Dateien/Lehre/Business\\_IT/WS\\_05-06/BIT-ws05-07SecurityBiometrie-Studie.pdf](http://www.wi3.uni-erlangen.de/fileadmin/Dateien/Lehre/Business_IT/WS_05-06/BIT-ws05-07SecurityBiometrie-Studie.pdf), 2003, P.10, retrieved 16.08.2008

Anagun A. S., Designing A Neural Network Based Computer Access Security System, Keystroke Dynamics and/or Voice Patterns. In, Smart Engineering System Design 4 (2002), Nr. 2, P. 125–132

Ashbourn, J., Biometrics – Advanced Identity Verification. London, Springer, 2000

Ashley, P., Wandenwauver, M. Practical intranet security, Overview of the state of the art and available technologies, Norwell, USA, 1999

Bakdi, I., Benutzerauthentifizierung anhand des Tippverhaltens bei Verwendung fester Eingabetexte, 2007

Bartmann, D., Bakdi, I., Achatz, M., On the Design of an Authentication System Based on Keystroke Dynamics Using a Predefined Input Text. In, International Journal of Information Security and Privacy 1 (2007), Nr. 2, P. 1–12

Bartmann, D., Bartmann, D. jun., Verfahren zur Verifizierung der Identität eines Benutzers einer mit einer Tastatur zur Erzeugung alphanumerischer Zeichen zu bedienenden Datenverarbeitungsanlage, Deutsches Patent, Nummer 196 31 484, 1997.

Bartmann, D., Benutzerauthentisierung durch Analyse des Tippverhaltens mit Hilfe einer Kombination aus statistischen und neuronalen Verfahren, Herbert Utz Verlag, München, ISBN 3-89675-836-5.

- Bartmann, D., Breu, C., Eignung des biometrischen Merkmals Tippverhalten zur Benutzerauthentisierung, in, Bartmann, D., Mertens, P., Sinz, E. J. (Hrsg.), Überbetriebliche Integration von Anwendungssystemen, FORWIN-Tagung 2004. Aachen 2004, P. 321-341.
- Bartmann, D., Verfahren zur Verifizierung der Identität eines Benutzers einer mit einer Tastatur zur Erzeugung alphanumerischer Zeichen zu bedienenden Datenverarbeitungsanlage. Europäische Patentschrift, Nummer EP 0 917678 B1, 2001
- Bartmann, D., Wimmer, M., Kein Problem mehr mit vergessenen Passwörtern, DuD Journal, 2007
- Behrens, M., Roth R. (Hrsg.), Biometrische Identifikation, Grundlagen, Verfahren, Konzepte, Vieweg und Teubner Verlag, Veröffentlicht 2001, [www.biometrics-institute.com/pdf/Grundlagen.pdf](http://www.biometrics-institute.com/pdf/Grundlagen.pdf)
- Betschart, W., Applying intelligent statistical methods on biometric systems, 2005
- Bhattacharjee, A. Individual Trust in Online Firms, Scale Development and Initial Test. Journal of Management Information Systems, 19 (1), 211-242, 2002
- BioAPI Consortium, [www.bioapi.org](http://www.bioapi.org), retrieved 01.10.2008
- Birch, D. G.W., Digital identity management, perspectives on the technological, business and social implications. Gower, Aldershot 2007
- Blöckenwegner, J., Aiglesberger, M., Identität, [www.stangl.eu/psychologie/definition/Identitaet.shtml](http://www.stangl.eu/psychologie/definition/Identitaet.shtml), 2006, retrieved 16.08.2008
- Breden, N. und Schröder, B., Projekt ESPRESSO, [www.informatik.uni-bremen.de/agbkb/lehre/espresso/files/referate/biometrie.pdf](http://www.informatik.uni-bremen.de/agbkb/lehre/espresso/files/referate/biometrie.pdf), retrieved 16.08.2008
- Breese, N., Password crack using a PS3 console, [news.bbc.co.uk/2/hi/technology/7118997.stm](http://news.bbc.co.uk/2/hi/technology/7118997.stm) 2007, retrieved 18.12.2007
- Breu, C., Bartmann, D., Authentisierung anhand des Tippverhaltens, [www.kes.info/archiv/online/01-04-46-TippAuthent.htm](http://www.kes.info/archiv/online/01-04-46-TippAuthent.htm), 2001, retrieved 16.08.2008
- Bromba, M., [www.bromba.com/faq/biofaqd.htm#Sample](http://www.bromba.com/faq/biofaqd.htm#Sample), 2008, retrieved 16.08.2008

Bundesamt für Sicherheit in der Informationstechnik, Biometrie,  
[www.bsi.de/fachthem/biometrie/index.htm](http://www.bsi.de/fachthem/biometrie/index.htm), retrieved 16.08.2008

Bundesamt für Sicherheit in der Informationstechnik, Webkurs GSTOOL – Glossar.  
[www.bsi.bund.de/gstool/wbtgstool/wbtgstool/kurs/glossar.htm](http://www.bsi.bund.de/gstool/wbtgstool/wbtgstool/kurs/glossar.htm), retrieved, 12.07.2008.

Bundeskriminalamt, [www.bka.de/pks/pks2007/p\\_3\\_21.pdf](http://www.bka.de/pks/pks2007/p_3_21.pdf), retrieved 01.08.2008

Bundestrojaner, [www.bundestrojaner.de](http://www.bundestrojaner.de), retrieved 01.09.2008

Burton Group Identity Blog, [identityblog.burtongroup.com](http://identityblog.burtongroup.com), retrieved 01.09.2008

Burton Group, Identity and Security.  
[www.burtongroup.com/consulting/PractiseIdentitySecurityServices.aspx](http://www.burtongroup.com/consulting/PractiseIdentitySecurityServices.aspx), retrieved 06.07.2008

Cameron, K., Integrating OpenID and InfoCard [www.identityblog.com/?p=659](http://www.identityblog.com/?p=659), retrieved  
11.8.2007

Cameron, K., The laws of identity. [www.identityblog.com/?p=352](http://www.identityblog.com/?p=352), 08.01.2006, 08.01.2006,  
retrieved 24.06.2008

Cantor, S. Shibboleth architecture protocols and profiles working draft 05,  
[shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-02.pdf](http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-02.pdf), 2004, retrieved  
01.05.2008

Captcha - Completely Automated Public Turing test to tell Computers and Humans Apart,  
[www.captcha.net](http://www.captcha.net), retrieved 12.08.2008

Central Authentication Service, [www.ja-sig.org/products/cas](http://www.ja-sig.org/products/cas), retrieved 01.05.2008

Centre for Mathematics and Scientific Computing am National Physical Laboratory in England,  
Comparison of different biometric methods, 2002

Chappel, D., What Is CardSpace? [blogs.msdn.com/irenak/archive/2006/08/23/714626.aspx](http://blogs.msdn.com/irenak/archive/2006/08/23/714626.aspx),  
2006, retrieved 12.08.2008

Cho, S., Han, C., Han, D., K., H., Web-based keystroke dynamics identity verification using neural  
networks, Journal of Organisational Computing & Electronic Commerce, 10(4), 295-307, 2001.

Compliance Magazin: Governance, Risk & Compliance, Die Bedeutung von Records Management für GRC. [www.compliancemagazin.de/markt/nachrichten/projectconsult140508.html](http://www.compliancemagazin.de/markt/nachrichten/projectconsult140508.html), 14.05.2008, retrieved 04.07.2008

Computerwelt, Wider die Passwort-Flut. [www.computerwelt.at/detailArticle.asp?a=107936&n=4](http://www.computerwelt.at/detailArticle.asp?a=107936&n=4), 2006, retrieved 06.09.2008

Cser, A., Penn, J., Identity Management Market Forecast, 2007 To 2014, Provisioning Will Extend Its Dominance Of Market Revenues. [www.forrester.com/Research/Document/Excerpt/0,7211,43842,00.html](http://www.forrester.com/Research/Document/Excerpt/0,7211,43842,00.html), 06.02.2008, retrieved 06.07.2008

Dawson, E. Lopez, J. Montenegro, J.A. Okamoto, E (2003), BAAI, biometric authentication and authorization infrastructure, International Conference on Information Technology, Research and Education, 2003

Diffie-Hellman algorithm, [www.rsa.com/rsalabs/node.asp?id=2248](http://www.rsa.com/rsalabs/node.asp?id=2248), retrieved 01.03.2008

DOT NET Framework Developer Center, Microsoft Corporation, Informationen über Informationskarten und digitale Identität. [msdn.microsoft.com/de-de/library/ms734655.aspx](http://msdn.microsoft.com/de-de/library/ms734655.aspx), 2007, retrieved, 08.08.2008

Doupnik, J., Notes about the design of an identity vault, Universität Regensburg, 2008

Eckert, C., IT-Sicherheit Konzepte - Verfahren - Protokolle, Oldenbourg Wissenschaftsverlag GmbH, 2008

Erber, R., Schläger, C., Pernul, G., Patterns for Authentication and Authorisation Infrastructures. Proc. of the 1st International Workshop on Secure Systems Methodologies using Patterns (SPattern'07), Regensburg, Germany, 2007.

Erickson, J., Hacking - The art of exploitation, No starch press, San Francisco, 2003

Fakos, A., Sichere Webanwendungen, Grundlagen, Schwachstellen und Gegenmaßnahmen. Vdm Verlag Dr. Müller, Saarbrücken 2007

Fernandez, E. B., Pernul, G., Larrando-Petrie, M., Patterns und Pattern Diagrams for Access Control. Proc. of the 5th International Conference on Trust, Privacy & Security in Digital Business (TrustBus '08), Italy, 2008

Forrester Group, [www.forrester.com](http://www.forrester.com), retrieved 01.09.2008

Friedmann, K., SOA-Initiativen schlampen bei Sicherheit.

[www.computerwoche.de/knowledge\\_center/security/1866843](http://www.computerwoche.de/knowledge_center/security/1866843), 18.06.2008, retrieved 25.06.2008

Geschonneck, A., Studie zu Angriffen von Innentätern, computer-

[forensik.org/2008/01/29/studie-zu-angriffen-von-innentatern](http://forensik.org/2008/01/29/studie-zu-angriffen-von-innentatern), 29.01.2008, retrieved 24.06.2008

Global Security, [www.globalsecurity.org/security/systems/images/biometric-intro.gif](http://www.globalsecurity.org/security/systems/images/biometric-intro.gif), 2008, retrieved 16.08.2008

Gunetti, D., Picardi, C., Keystroke analysis of free text, ACM Transactions On Information and System Security, 3(5), 312-347, 2005.

Hansen, M., Meints, M., Digitale Identitäten — Überblick und aktuelle Trends, Identity-Lifecycle, Authentisierung und Identitätsmanagement, Datenschutz und Datensicherheit – DuD.

[www.fidis.net/fileadmin/fidis/publications/2006/DuD09\\_2006\\_543.pdf](http://www.fidis.net/fileadmin/fidis/publications/2006/DuD09_2006_543.pdf), Volume 30 Nummer 9, 2006, retrieved 24.07.2008

Hardt, D., Bufo, J et al., OpenID Attribute Exchange 1.0 – Final, [openid.net/specs/openid-attribute-exchange-1\\_0.html](http://openid.net/specs/openid-attribute-exchange-1_0.html) , retrieved 17.8.2008

Hardt, D., Identity 2.0, What is user-centric identity? [identity20.com/?p=61](http://identity20.com/?p=61), 26.06.2006, retrieved 08.08.2008

Hardt, D., Identity 2.0. [identity20.com/media/OSCON2005/](http://identity20.com/media/OSCON2005/), 2005, retrieved 15.08.2008

Hardt, D., Sxip, [http://www.sxip.com/dick\\_hardt](http://www.sxip.com/dick_hardt), retrieved 15.08.2008

Hauptvogel, K., Ritzschke, M., Biometrie um die Jahrhundertwende, 2004

Hess, A., Humm, B., Voß, M., Regeln für serviceorientierte Architekturen hoher Qualität, Springer-Verlag, 2006.

Hommel, W., Architektur- und Werkzeugkonzepte für föderiertes Identitäts-Management, Dissertation, München 2007

Hymér, P., Extraction and Application of Secondary Crease Information in Fingerprint Recognition Systems, 2005

Internet2 consortium, [www.internet2.edu](http://www.internet2.edu), retrieved 15.09.2008

ISACA Group, Document G36 - Biometric Controls, [www.isaca.org/Template.cfm?Section=home&Template=/ContentManagement/ContentDisplay.cfm&ContentID=27712](http://www.isaca.org/Template.cfm?Section=home&Template=/ContentManagement/ContentDisplay.cfm&ContentID=27712), retrieved 14.08.2008

IT Reseller, Biometrie wird wiederbelebt. [itreseller.ch/archive/ar\\_single.cfm?ar\\_id=25074&sid=0](http://itreseller.ch/archive/ar_single.cfm?ar_id=25074&sid=0), 30.06.2008, retrieved 12.08.2008

ITLexikon, Biometrie, [www.itwissen.info/definition/lexikon/Biometrie-biometrics.html](http://www.itwissen.info/definition/lexikon/Biometrie-biometrics.html), 2008, retrieved 16.08.2008

ITWissen, SOA (service oriented architecture). [www.itwissen.info/definition/lexikon/service-oriented-architecture-SOA-SOA-Architektur.html](http://www.itwissen.info/definition/lexikon/service-oriented-architecture-SOA-SOA-Architektur.html), retrieved 04.07.2008

ITWissen, [www.itwissen.info/definition/lexikon/equal-error-rate-EER.html](http://www.itwissen.info/definition/lexikon/equal-error-rate-EER.html), 2008, retrieved 16.08.2008

Jain, A.K. Biometric recognition, how do I know who you are? Signal Processing and Communications Applications Conference, 2004

Jain, A.K., Ross, A., Prabhakar, S., Circuits and Systems for Video Technology, IEEE Transactions on Volume 14, Issue 1, 2004, P. 4-20

JanRain, OpenID stats on May 1st 2008, [janrain.com/blog/2008/05/01/openid-stats-on-may-1st-2008](http://janrain.com/blog/2008/05/01/openid-stats-on-may-1st-2008), 2008, retrieved 28.06.2008

Jendricke, U., Sichere Kommunikation zum Schutz der Privatsphäre durch Identitätsmanagement. Rhombos-Verlag, Berlin 2003

Jerra Soft, Fingerprint Reader / Scanner - Fingerabdruck-Erkennung, [www.biometrie-identifikation.de/fingerprint.php](http://www.biometrie-identifikation.de/fingerprint.php), 2008, retrieved 16.08.2008

Joyce, R., Gupta, G., , Identity Authentication Based on Keystroke Latencies. In, Communications of the ACM 33 (1990), Nr. 2, P. 168–176

KeyloggerPro.com, [www.keyloggerpro.com](http://www.keyloggerpro.com), retrieved 01.09.2008

Kholmatov A., Yanikoglu, B., Biometric Cryptosystem Using Online Signatures, Computer and Information Sciences – ISCIS 2006

Klünter, D., Laser, J., LDAP verstehen, OpenLDAP einsetzen, Grundlagen und Praxiseinsatz. dpunkt.verlag, Heidelberg 2008

Kontani, K., Horii, K., Evaluation on a keystroke authentication system by keying force incorporated with temporal characteristics of keystroke dynamics. In, Behaviour & Information Technology 24 (2005), Nr. 4, P. 289–302

Korman, D., Rubin A., Risks of the Passport Single Signon Protocol, Computer Networks, Elsevier Science Press, volume 33, pages 51-58, 2000.

Kowal, J., Rollenbasierte Sicherheitsmodelle. [www2.informatik.hu-berlin.de/Forschung\\_Lehre/algorithmenII/Lehre/WS2003-2004/Sem\\_Security/05RBAC/RBAC\\_paper.pdf](http://www2.informatik.hu-berlin.de/Forschung_Lehre/algorithmenII/Lehre/WS2003-2004/Sem_Security/05RBAC/RBAC_paper.pdf), retrieved 08.08.2008

Krafzig, D., Banke, K., Slama, D., Enterprise SOA, Prentice Hall Professional Technical Reference, 2005

Kuppinger, M., Analystengruppe KCP nennt die 10 Top-Trends im Identity Management. [www.kuppingercole.com/topstory/05.03.2008](http://www.kuppingercole.com/topstory/05.03.2008), 05.03.2008, retrieved 04.07.2008

Kuppinger, M., GRC and IAM – you can't separate it, [blogs.kuppingercole.com/kuppinger/2008/06/06/grc-and-iam-you-cant-separate-it](http://blogs.kuppingercole.com/kuppinger/2008/06/06/grc-and-iam-you-cant-separate-it), 06.06.2008, retrieved 04.07.2008

Kuppinger, M., How shall a GRC solution look like? [blogs.kuppingercole.com/kuppinger/2007/10/31/how-shall-a-grc-solution-look-like](http://blogs.kuppingercole.com/kuppinger/2007/10/31/how-shall-a-grc-solution-look-like), 31.10.2007, retrieved 04.07.2008

Kuppinger, M., Kerntechnologien des Identity Management, Übersicht über die technischen Komponenten des Identity Management. [www.kuppingercole.com/articles/im\\_kerntechnologien](http://www.kuppingercole.com/articles/im_kerntechnologien), 10.01.2004, retrieved 29.06.2008

Kuppinger, M., Monolithische und überfrachtete Identity-Suiten sind der falsche Weg. [www.computerzeitung.de/articles/monolithische\\_und\\_ueberfrachtete\\_identity-suiten\\_sind\\_der\\_falsche\\_weg](http://www.computerzeitung.de/articles/monolithische_und_ueberfrachtete_identity-suiten_sind_der_falsche_weg),

/2008011/31428357\_ha\_CZ.html?path=/articles/2008011/31428357\_ha\_CZ.html&art=/articles/2008011/31428357\_ha\_CZ.html&thes=&pid=ee54f3c7-0de1-40f5-bb23-2cfd022aee5, 07.03.2008, retrieved 04.07.2008

Kuppinger, M., The secret leader in context-based authentication and authorization? [blogs.kuppingercole.com/kuppinger/category/context-based-authorization](http://blogs.kuppingercole.com/kuppinger/category/context-based-authorization), 19.06.2008, retrieved 12.08.2008

Laßmann, Dr., Bewertungskriterien zur Vergleichbarkeit biometrischer Verfahren, [www.teletrust.de/fileadmin/files/publikationen/KritKat-3\\_final.pdf](http://www.teletrust.de/fileadmin/files/publikationen/KritKat-3_final.pdf), 2006, S.6-7, retrieved 16.08.2008

Levenstein distance algorithm, [www.levenshtein.de](http://www.levenshtein.de), retrieved 01.07.2008

Liberty Alliance, Introduction to the Liberty Alliance Identity Architecture, [xml.coverpages.org/LibertyAllianceArchitecture200303.pdf](http://xml.coverpages.org/LibertyAllianceArchitecture200303.pdf), retrieved 20.7.2008

Livejournal, [www.livejournalinc.com](http://www.livejournalinc.com), 2008, retrieved 02.09.2008

Lopez, J., Oppliger, R., Pernul, G., Authentication and authorisation infrastructures (AAIs), a comparative survey, *Computers&Security* 23, 578-590, 2004

Makezine.com, [blog.makezine.com/archive/electronics/121.html](http://blog.makezine.com/archive/electronics/121.html), retrieved 14.08.2008

Mayer, R.C., Davis, J.H., Schoorman, F.D. (1995), An Integrative Model of Organizational Trust. *Academy of Management Review*, 20 (3), 709-734

Mcknight, D.H., Cummings, L.L. e Chervany, N.L. [1998], Initial Trust Formation in New Organizational Relationships, in «Academy of Management Review», vol. 23, n. 3

Meints, M., Identität, Vieweg Verlag, Datenschutz und Datensicherheit – DuD, [springerlink.com/content/wp6806l206t6774l/fulltext.pdf](http://springerlink.com/content/wp6806l206t6774l/fulltext.pdf), Volume 30 Nummer 6, September 2006, retrieved 24.07.2008

Mertens, P. (Hrsg.), Sinz, E.. (Hrsg.), Überbetriebliche Integration von Anwendungssystemen – FORWIN-Tagung 2004. Aachen, Shaker Verlag, S. 321–341, 2004

Mezler-Andelberg, C., Identity Management – eine Einführung, Grundlagen, Technik, wirtschaftlicher Nutzen. dpunkt.verlag, Heidelberg 2008

Microsoft Passport, [www.passport.net](http://www.passport.net), retrieved 01.10.2008

Microsoft supports OpenID, [www.microsoft.com/presspass/press/2007/feb07/02-06RSA07KeynotePR.msp](http://www.microsoft.com/presspass/press/2007/feb07/02-06RSA07KeynotePR.msp), retrieved 01.02.2008

Microsoft, Verwenden von CardSpace in Windows Communication Foundation. [msdn.microsoft.com/de-de/library/ms733090.aspx](http://msdn.microsoft.com/de-de/library/ms733090.aspx), 2007, retrieved 02.09.2008

Mitsubishi Electric Research Laboratories, [www.merl.com/projects/irisrecognition](http://www.merl.com/projects/irisrecognition), 2007, retrieved 16.08.2008

Monrose, F., Reiter, M., Wetzel, S., Password hardening based on keystroke dynamics, *International Journal of Information Security*, 1(2), 2001

Mut-Puigserver M., Payeras-Capellà M., Ferrer-Gomila J. and Huguet-Rotger L., Replay Attack in a Fair Exchange Protocol, *Applied Cryptography and Network Security*, Volume 5037, 2008

Nanda, A., Bhargava, Ruchi, CardSpace – Deep Architecture. [channel9.msdn.com/Showpost.aspx?postid=192473](http://channel9.msdn.com/Showpost.aspx?postid=192473) (Webcast), 12.05.2006, retrieved, 08.08.2008

National Institute of Standards and Technology, Role Based Access Control (RBAC) and Role Based Security. [csrc.nist.gov/groups/SNS/rbac/index.html](http://csrc.nist.gov/groups/SNS/rbac/index.html), April 2008, retrieved 29.06.2008

Needleman S., Wunsch D., A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48, 443–453, 1970

Neher, M., Compliance und Identity Management, [blog.doubleslash.de/2006/10/30/compliance-und-identity-management](http://blog.doubleslash.de/2006/10/30/compliance-und-identity-management), 30.10.2006, retrieved 30.07.2008

Neher, M., IAM & SOA - Zwangsheirat oder Liebeshochzeit, [blog.doubleslash.de/2006/11/16/iam-soa-zwangsheirat-oder-liebeshochzeit](http://blog.doubleslash.de/2006/11/16/iam-soa-zwangsheirat-oder-liebeshochzeit), 16.11.2006, retrieved 04.07.2008

Net-Report, Meeting the Data Management Challenges for Basel II Readiness with Net Report. [www.symbolic.it/prodotti/net\\_report/download/Basilea\\_II\\_NetReport.pdf](http://www.symbolic.it/prodotti/net_report/download/Basilea_II_NetReport.pdf), 2005, retrieved 08.08.2008

Networld, Yahoo wird OpenID-fähig. [www.golem.de/0801/57095.html](http://www.golem.de/0801/57095.html), 2008, retrieved 02.09.2008

NIST American National Institute of Standards and Technology, [www.nist.gov](http://www.nist.gov), retrieved 15.09.2008

NIST eHandbook of statistical methods, <http://www.itl.nist.gov/div898/handbook/>, retrieved 15.09.2008

Nolde, V., Biometrische Verfahren – Körpermerkmale als Passwort, 2002

Olden, M, Za, S., Federated AAIs with biometric authentication, Increasing security and trust in organizational relationships, itAIS 2008, to be published

Olden, M., Problems of biometric methods in authentication and authorisation infrastructures, Scientifical Journal of the University of Pitesti, 2008, to be published

Omote, K., Okamoto, E., User Identification System Based on Biometrics for Keystroke, 1999, P.216-229

OpenID Developers Specifications, [openid.net/developers/specs](http://openid.net/developers/specs), retrieved 05.05.2008

OpenID Authentication 2.0 - Final, [openid.net/specs/openid-authentication-2\\_0.html](http://openid.net/specs/openid-authentication-2_0.html), retrieved 28.7.2008

OpenID Authentication 2007, [www.openid.net](http://www.openid.net), retrieved 01.03.2007

OpenID Blog Germany, OpenID History, 2007, [openidgermany.de/openid-historie/](http://openidgermany.de/openid-historie/), retrieved 11.07.2008

OpenID Blog Germany, Wie funktioniert OpenID?, 2007, [openidgermany.de/openid-basiswissen/wie-funktioniert-openid/](http://openidgermany.de/openid-basiswissen/wie-funktioniert-openid/), retrieved 11.07.08

OpenID Community Wiki, Terminology, 2008, [wiki.openid.net/Terminology](http://wiki.openid.net/Terminology), retrieved 11.07.2008

OpenID Europe, History of OpenID. [www.openideurope.eu/openid/history](http://www.openideurope.eu/openid/history), 2008, retrieved 02.09.2008

OpenID PHP API Documentation, [openidenabled.com/files/php-openid/docs/2.1.1](http://openidenabled.com/files/php-openid/docs/2.1.1), retrieved 9.8.2007

OpenID Specifications, [specs.openid.net/auth/2.0/identifier\\_select](http://specs.openid.net/auth/2.0/identifier_select), retrieved 01.05.2008

OpenID Spread Technical Terms, 2008, [spreadopenid.org/technical-terms/](http://spreadopenid.org/technical-terms/), retrieved 11.07.2008

OpenID, OpenID Foundation, [openid.net/foundation](http://openid.net/foundation), 2008, retrieved 02.09.2008

OpenID, [openidenabled.com/php-openid](http://openidenabled.com/php-openid), 2008, retrieved 10.05.2008

Paasche, L., Brüning, W., Geschichte der Biometrie, Otto-von-Guericke-Universität Magdeburg, 2005

PAPE Policies, [schemas.openid.net/pape/policies](http://schemas.openid.net/pape/policies), retrieved 01.05.2008

Parthier, U., eProvisioning & Identity Management, Business Value, Markt & Anbieter. IT Research, Sauerlach 2003

Pavlou, P., Tan, Y.H. and Gefen, D., Institutional Trust and Familiarity in Online Interorganizational Relationship, 2003

Peacock, A., Ke, X., Identifying Users from their Typing Patterns, O'Reilly 2005, P.199-200

Peacock, A., Ke, X., Wilkerson, M., Typing Patterns, A Key to User Identification, IEEE Security and Privacy, No. 5, Vol. 2, (2004) 40-47

Pear, Package Information DB, [pear.php.net/package/DB](http://pear.php.net/package/DB), 2008, retrieved 30.08.2008

Pernul, G., Schläger, C., Muschall, B., Implementierung einer kompletten Liberty Infrastruktur, University of Regensburg, 2006

Pernul, G., Unland, R., Datenbanken im Unternehmen 2nd edition, Oldenbourg Wissenschaftsverlag, München, 2003.

PINYIN, [www.pinyin.info](http://www.pinyin.info), retrieved 01.10.2008

Psylock - biometric authentication by typing behaviour, [www.psylock.de](http://www.psylock.de), retrieved 01.09.2008

Pugliese J., Biometrics, Infrastructural Whiteness, and the Racialized Zero Degree of Nonrepresentation, Duke University Press, 2007

Qarchive, Database synchronisation, [database-synchronization.qarchive.org](http://database-synchronization.qarchive.org), retrieved 01.10.2008

Radicati Group, Identity Management Market Grows Rapidly, Driven by Regulations and Security Issues. [findarticles.com/p/articles/mi\\_pwwi/is\\_200702/ai\\_n17219693/pg\\_1](http://findarticles.com/p/articles/mi_pwwi/is_200702/ai_n17219693/pg_1), Februar 2007, retrieved 06.07.2008

Ratha, N. K., Connell, J. H., Bolle, R. M., Enhancing security and privacy in biometrics-based authentication systems, IBM Systems Journal, Volume 40, Number 3, 2001

Rehman, Rafeeq, The OpenID Book, [www.openidbook.com](http://www.openidbook.com), retrieved 4.7.2007

Richter, M., Identity Management, Integration der Benutzerverwaltung in einer heterogenen Systemlandschaft. Vdm Verlag Dr. Müller, Saarbrücken 2007.

Rohr, S., Governance, Risk, Compliance (GRC) & SOA Identity Management. [www.sap-im-betrieblichen-spannungsfeld.de/material/2\\_tag/ws08sebastianrohr\\_soa\\_sicherheit.pdf](http://www.sap-im-betrieblichen-spannungsfeld.de/material/2_tag/ws08sebastianrohr_soa_sicherheit.pdf), 14.02.2008, retrieved 04.07.2008

Rosenberg, D., Top 10 trends in identity management. [news.cnet.com/8301-13846\\_3-9889837-62.html](http://news.cnet.com/8301-13846_3-9889837-62.html), 10.03.2008, retrieved 04.07.2008

RSA Security Study about password management, [www.rsasecurity.com/passwords](http://www.rsasecurity.com/passwords), 2008

Rubin, A., Keystroke Dynamics as a Biometric for Authentication, Elsevier Preprint, 1999

Rummeyer, O., Düsterhaus, J., SSO frei Haus, Java Magazin, 10.2006

Schema for OpenID attribute exchange, [axschema.org](http://axschema.org), retrieved 01.10.2008

Scherrbacher, K., IdM, Die gängigsten Irrtümer, [www.computerwoche.de/knowledge\\_center/it\\_security/594470](http://www.computerwoche.de/knowledge_center/it_security/594470), 20.06.2007, retrieved 02.07.2008

Schläger, C., Attribute-based Infrastructures for Authentication and Authorisation, EUL Verlag 2008 – Reihe Electronic Commerce, Band 36, ISBN, 978-3-89936-670-9

Schläger, C., Ganslmayer, M., Effects of Architectural Decisions in Authentication and Authorisation Infrastructures. Proc. of the 2nd International Conference on Availability, Reliability and Security (ARES '07), Vienna, 2007

Schläger, C., Sojer, M., Muschall, .B., Pernul, G., Attribute-Based Authentication and Authorisation Infrastructures for E-Commerce Providers, pp132-141 Springer-Verlag, 2006

Seidel, Bernd, Starker SOA-Einsatz bringt Probleme. [www.zdnet.de/itmanager/strategie/0,39023331,39188349,00.htm](http://www.zdnet.de/itmanager/strategie/0,39023331,39188349,00.htm), 13.03.2008, retrieved 04.07.2008

Sheng, Y., Phoha, V.V., Rovnyak, S.M., A parallel decision tree-based method for user authentication based on keystroke patterns, in IEEE Transactions on Systems, Man, and Cybernetics, Part B, 35(4), 826-833, 2005

Siemens, Siemens bündelt Identity Management und Biometrie. <https://www.it-solutions.siemens.com/b2b/it/de/global/presse/pressemeldungen/2008/Pages/identity-management-biometrie.aspx>, 08.05.2008, retrieved 05.07.2008

Smarty, Is Smarty right for me. [www.smarty.net/rightforme.php](http://www.smarty.net/rightforme.php), 2008, retrieved 02.09.2008

Smith, R.E., Authentication, From password to public keys, Addison-Wesley Longman, Amsterdam, 2002, P. 448

Spagnoletti P., Za S., D'Atri A. Institutional Trust and security, new boundaries for Virtual Enterprises, to appear in Proc. of 2nd International Workshop on Interoperability Solutions to Trust, Security, Policies and QoS for Enhanced Enterprise Systems, IS-TSPQ2007, March 26th, 2007 Funchal (Madeira Island), Portugal

Spagnoletti P., Za S., Identity management in a cross boarder e-services environment, the LD-CAST case, to appear in Proc. 6th Eastern European eGovernment Days, 23 - 25, Prague 2008

Spie, [spie.org/Images/Graphics/Newsroom/Imported/0815/0815\\_fig2.jpg](http://spie.org/Images/Graphics/Newsroom/Imported/0815/0815_fig2.jpg), 2008, retrieved 16.08.2008

Spiegel, [www.spiegel.de/netzwelt/tech/0,1518,288462,00.html](http://www.spiegel.de/netzwelt/tech/0,1518,288462,00.html), 2004, retrieved 16.08.2008

Spillane, R. J., Keyboard apparatus for personal identification, IBM Technical Disclosure Bulletin, Bd. 17, Nr. 11, 1975

Stein, L., Stewart, J., WWW Security FAQ, [www.w3.org/Security/Faq/wwwsf1.html](http://www.w3.org/Security/Faq/wwwsf1.html), 2003

Sulzberger, C., [www.levenshtein.de](http://www.levenshtein.de), retrieved 16.08.2008

Sun Microsystems used OpenID, [www.sun.com/aboutsun/pr/2007-05/sunflash.20070507.4.xml](http://www.sun.com/aboutsun/pr/2007-05/sunflash.20070507.4.xml), 2007

- Sury, U., Identity-Management und Recht, Identity-Management-Systeme. Springer Berlin/Heidelberg, Informatik-Spektrum, [www.advokatinnen.ch/html/publikationen/pfd/infospektrum304Identity.pdf](http://www.advokatinnen.ch/html/publikationen/pfd/infospektrum304Identity.pdf), Vol. 27 Nr. 3, 16.04.2004, retrieved 30.07.2008
- SWITCH, Swiss Education & Research Network [www.switch.ch/aai/demo/medium.html](http://www.switch.ch/aai/demo/medium.html), retrieved 11.8.2007
- Sxip, [www.sxip.com](http://www.sxip.com), retrieved 01.05.2008
- Syverson, P., A Taxonomy of Replay Attacks, Proceedings of the Computer Security Foundations Workshop VII, Franconia NH, 1994, IEEE CS Press (Los Alamitos, 1994).
- Tapiador, M., Fuzzy Keystroke Biometrics on Web Security, 1999, S.133-136
- Todorov, D., Mechanics of user identification and authentication, Fundamentals of Identity Management. Auerbach Publications, Boca Raton 2007.
- Turner, C.W., How consumers form their judgement of the security of e-commerce web-sites?, Workshop on Human-Computer Interaction and Security Systems, CHI2003, 2003
- Vaishnavi V. and Kuechler W. Design Research in Information Systems, July 2004 [www.isworld.org/Researchdesign/drisISworld.htm](http://www.isworld.org/Researchdesign/drisISworld.htm) Retrieved 01.10.2008
- Verisign, Fact Sheet. [www.verisign.com/corporate/fact-sheet/index.html](http://www.verisign.com/corporate/fact-sheet/index.html), retrieved 02.09.2008
- Verisign, VeriSign tritt OpenID Foundation bei. [www.verisign.de/verisign-inc/page\\_039422.html](http://www.verisign.de/verisign-inc/page_039422.html), retrieved 02.09.2008
- Walser, M., Compliance, Pflicht oder Kür für den IT-Leiter? [www.inside-it.ch/frontend/insideit?&site=ii&\\_d=\\_article&news.id=8537](http://www.inside-it.ch/frontend/insideit?&site=ii&_d=_article&news.id=8537), 11.10.2006, retrieved 11.07.2008
- Walther, H., Kein vernetztes Arbeiten ohne digitale Identität. [www.computerwoche.de/knowledge\\_center/it\\_security/594015](http://www.computerwoche.de/knowledge_center/it_security/594015), 11.06.2007, retrieved 27.06.2008
- Wikipedia - Die freie Enzyklopädie, [de.wikipedia.org/wiki/Informationssicherheit#Angriffe\\_und\\_Schutz](http://de.wikipedia.org/wiki/Informationssicherheit#Angriffe_und_Schutz), retrieved 08.06.2008

Wohlgemuth, Sven et al., Sicherheit und Benutzbarkeit durch Identitätsmanagement. [www.iig.uni-freiburg.de/telematik/atus/publications/WoJeGeDoMu2004.pdf](http://www.iig.uni-freiburg.de/telematik/atus/publications/WoJeGeDoMu2004.pdf), 2004, retrieved 12.08.2008

Ybarra, D., WISEKey, Microsoft and the OISTE Foundation cooperates to reduce the gap in the new Digital Identification Divide by issuing 20 million Digital Identities at WSIS.  
[www.presseportal.ch/de/pm/100006027/100500025/wisekey\\_sa?pre=1](http://www.presseportal.ch/de/pm/100006027/100500025/wisekey_sa?pre=1), 16.11.2005, retrieved 24.06.2008

Ying-Han P., Andrew T., Two-Factor Cancelable Biometrics Authenticator, Journal of Computer Science and Technology, Volume 22, Number 1, 2007

Zaytsev, O., Rootkits, Spyware/Adware, Keyloggers and backdoors - detection and neutralization, A-LIST, LLC, First Edition, 2007.

