On Petri Net Representations of Cryptographic Workflows in Electronic Government

Peter Lory
University of Regensburg
D-93040 Regensburg, Germany
Peter.Lory@wiwi.uni-regensburg.de

Abstract

Cryptographic workflows are a concept with a high potential for electronic government. It has been developed in the literature on identity based cryptography. It allows a government as the issuer of an electronic document to enforce the recipient to carry out certain actions before the latter can read the document (policy enforcement). The present paper models the underlying processes by Petri nets and extracts the common features of these cryptographic workflow nets. It shows that they have a series of attractive properties. Further, the paper focuses on the fact that the complete workflow net is composed of a net PN_1, which models the production of the unencrypted document, and the cryptographic workflow net PN_2. It demonstrates that desirable properties of the Petri net PN_1 are inherited by the composed net. Hence, the incorporation of policy enforcements does not introduce constructs that would complicate the treatment of the arising Petri nets.

1. Introduction

The phrase *cryptographic workflow* was coined by Al-Riyami and Paterson [6] although the underlying idea has already been used by Paterson [16], Chen et al. [10] and Smart [20]. Cryptographic workflows heavily employ identity based cryptography (see Boneh and Franklin [9]) or variants of it. Encryption schemes in this type of cryptography are characterized by the application of a user's public key, which can be calculated directly from a string representing the user's identity. This contrasts favorably with classical public key encryption, where the public key has to be extracted from a certificate issued by a certification authority in a public key infrastructure. Let, for example, the owner of a car with registration number `X675` apply for a document, which is issued by the government and requires the proof that this car has been insured. In an online dialogue with the owner, the government could establish the document and send it to the owner in encrypted form. The government can derive this public encryption key directly from the identity `X675`. The owner as the recipient of this ciphertext must then take care to come into possession of the corresponding private decryption key. For this purpose, the owner must contact the insurance company. Here the insurance company acts as a trusted authority and delivers this private key only, if the car with registration number `X675` is insured. In this way, the government as the issuer of the encrypted document can enforce the recipient to perform certain operations in order to achieve the decryption of the document. This feature is called *policy enforcement*. That part of the process that is enforced by the need to decrypt the document is named *cryptographic workflow*. Obviously, this concept allows interesting applications in electronic government.

A second example assumes a scenario of a press release, where a governmental text needs to be released simultaneously at 12.00 on June 15. A string describing this moment would be `1200:15:06`. In this case, the public key for the encryption of the text is directly calculated from this string. Again, the government knows the public key by simply knowing the identifier and no public key infrastructure is necessary. The fact that the text is encrypted enforces again certain actions by each of the recipients, who has to wait until the trust authority broadcasts the secret decryption key and then must decrypt the ciphertext. For this purpose, it is assumed that the trust authority is a time signal service, which timely broadcasts the secret keys corresponding to certain time steps (every 15 minutes say). In particular this service would broadcast the secret key corresponding to the string `1200:15:06` exactly at twelve o'clock of June 15. In this way it is guaranteed that each of the recipients gets the text at the same time. Due to applications like this it is preferred to talk about *identifier* based cryptography rather than *identity* based cryptography. The present paper also follows this custom. This is because the word identity implies the name of an entity (such as a user's email address or a car's registration number), while many applications actually associate keys with strings. Each string is able to represent anything, such as legal terms and conditions, and not just an entity's identity. Such a string describes what condition should be satisfied before the appropriate secret key is issued by the respective trust (or authorisation) authority. Such a secret key is often called a *credential* (see Al-Riyami, Malone-Lee and Smart [5] and Barbosa and Farshim [7]).

The present paper scans the current literature on cryptographic workflows and investigates the structure of the processes that are associated with them. For reasons outlined e. g. by van der Aalst [1] Petri nets are chosen to model these processes. Because these nets are graphical, they are easily accessible and easy to use. They also have a strong mathematical basis and there are elaborate analytical techniques available for them (see e. g. Verbeek, Basten and van der Aalst [22]). The present paper focuses on the common features that characterize Petri net representations of cryptographic workflows and points out that these workflow nets are nested (one enclosed in the other) or disjoint pairs of AND-splits/AND-joins and OR-splits/OR-joins respectively. It can be shown that they have attractive properties such as soundness (cf. van der Aalst and van Hee [4]).

In practice, the complete workflow net is composed of a net PN_1, which models the production of the unencrypted document, and the cryptographic workflow net. The application of the suggested techniques would be severely hindered, if the handling of the extended net is more difficult than that of the original net. The present paper shows that this is not the case. It investigates, how properties of the Petri net PN_1 are inherited by the composed net. It demonstrates that for usual desirable properties, in particular for soundness, this question can be answered positively. The special structure of cryptographic workflows allows stronger results than the general compositionality theorems reported in the literature (van der Aalst [3]).

The paper is organized as follows: Section 2 summarizes the basic principles of identifier based cryptography. Section 3 gives an example for a cryptographic workflow and extracts the common features in Petri net representations of cryptographic workflows. In Section 4 desirable properties of these workflow nets are proven and the inheritance of properties by the composed workflow net is investigated. Finally, the Conclusion interprets the results and gives an outlook for potential future work.

2. Principles of Identifier Based Cryptography

The history of identifier based cryptography dates back to 1984, when Shamir [19] introduced its concept and demonstrated that the authenticity problem in public key cryptography can be solved without the use of certification. An efficient solution for the corresponding confidentiality problem was given by Boneh and Franklin [9]. They employ the Weil pairing. There exists also a modification based on the Tate pairing [13], which is computationally less complex. This version will be outlined in the following two subsections for ease of reference (see e. g. Chen et al. [10]).

2.1. The Tate Pairing

Let G_1 and G_2 be two groups of prime order q in which the discrete logarithm problem is believed to be hard. Additionally, let t be an efficiently computable bilinear map

$$t : G_1 \times G_1 \to G_2 \,.$$

The operation in G_1 is written additively, since in applications G_1 will be the group of points on an elliptic curve. The group G_2 will denote a subgroup of the multiplicative group of a finite field and is written with a multiplicative notation. Consequently,

$$\begin{aligned} t(P_1 + P_2, Q) &= t(P_1, Q) \cdot t(P_2, Q)\,, \\ t(\lambda P, Q) &= t(P, Q)^\lambda\,, \\ t(P, Q_1 + Q_2) &= t(P, Q_1) \cdot t(P, Q_2)\,, \\ t(P, \mu Q) &= t(P, Q)^\mu \end{aligned}$$

for all $P, Q, P_1, P_2, Q_1, Q_2 \in G_1$ and $\lambda, \mu \in \mathbb{Z}_q$. The bilinearity of the map t allows a series of useful tricks. For example for $P, Q \in G_1$ and $\kappa, \lambda, \mu \in \mathbb{Z}_q$ we have

$$\begin{aligned} t(\kappa P, \lambda Q)^\mu &= t(\kappa P, \mu Q)^\lambda = t(\lambda P, \mu Q)^\kappa \\ &= t(\lambda P, \kappa Q)^\mu = t(\mu P, \kappa Q)^\lambda \\ &= t(\kappa\lambda P, Q)^\mu = t(\kappa\lambda P, \mu Q) \\ &= t(P, \kappa\lambda Q)^\mu = t(\mu P, \kappa\lambda Q) \\ &= \ldots \\ &= t(\kappa\lambda\mu P, Q) = t(P, \kappa\lambda\mu Q) \\ &= t(P, Q)^{\kappa\lambda\mu}\,. \end{aligned}$$

Further, two cryptographic hash functions are defined:

$$\begin{aligned} H_1 &: \{0,1\}^* \to G_1\,, \\ H_2 &: G_2 \to \{0,1\}^*\,. \end{aligned}$$

2.2. Identifier Based Encryption

There is some trust authority (TA), which owns a public/private key pair. This is a pair $(R_{\mathtt{TA}}, s)$ where $R_{\mathtt{TA}} \in G_1$ and $s \in \mathbb{Z}_q$ with

$$R_{\mathtt{TA}} = sP$$

for some given fixed point $P \in G_1$. The private (secret) key s of the trust authority is often called the 'master key'. An identifier based key pair is a pair $(Q_{\mathtt{ID}}, S_{\mathtt{ID}})$ where $Q_{\mathtt{ID}}, S_{\mathtt{ID}} \in G_1$ and the master key and the identifier based key pair are linked by

$$S_{\mathtt{ID}} = sQ_{\mathtt{ID}} \text{ and } Q_{\mathtt{ID}} = H_1(\mathtt{ID})\,.$$

Here, $\mathtt{ID}$ is the identifier string. The (identifier based) public key $Q_{\mathtt{ID}}$ can be directly derived from the identifier. It should be clearly distinguished between the private key s of the TA (the master key) and the identifier based private key $S_{\mathtt{ID}}$. Please note that $S_{\mathtt{ID}}$ can be generated only by the TA. For this reason, the TA is often (see e. g. [6]) called private key generator (PKG).

As usual in public key cryptography the sender encrypts a message by the public key of the addressee. Let the string $m \in \{0,1\}^*$ be the message to be encrypted. Then the identifier based encryption scheme of Boneh and Franklin [9] works as follows:

- **Encryption:**
  Compute $U = rP$ where r is a random element of $\mathbb{Z}_q$. Then compute
  $$V = m \oplus H_2(t(R_{\mathtt{TA}}, rQ_{\mathtt{ID}}))\,,$$
  where $\oplus$ denotes the bitwise addition modulo 2 (equivalent to the XOR-operation). The pair (U, V) is the ciphertext.

- **Decryption:**
  Decryption is performed by computing
  $$\begin{aligned} & V \oplus H_2(t(U, S_{\mathtt{ID}})) \\ &\quad = \quad V \oplus H_2(t(rP, sQ_{\mathtt{ID}})) \\ &\quad = \quad V \oplus H_2(t(P, Q_{\mathtt{ID}})^{rs}) \\ &\quad = \quad V \oplus H_2(t(sP, rQ_{\mathtt{ID}})) \\ &\quad = \quad V \oplus H_2(t(R_{\mathtt{TA}}, rQ_{\mathtt{ID}})) \\ &\quad = \quad m\,. \end{aligned}$$

Clearly, it has to be assumed that the recipient of the ciphertext owns the identifier based secret key $S_{\mathtt{ID}}$. Please note that the TA can always decrypt, too. This is the so called *key escrow property*. In some applications, such as disaster recovery, this escrow facility may be useful. However, for many applications this escrow property is undesirable and more sophisticated techniques have to be used (see Subsection 3.3).

3. Cryptographic Workflows

3.1. An Example

In this subsection a (slightly modified) example of Chen at al. [10] for a cryptographic workflow is presented in order to give an introduction into the basic concept. For details the reader is referred to this paper. The example extends the approach of Subsection 2.2 as it assumes that there are not only one but two trust authorities each with their own public/private key pair. It describes a possible online car tax disk dispenser in the United Kingdom. In this country every car needs to display a tax disk. This is purchased each year for a nominal fee, and essentially proves that at a given point in the year the owner of the car had car insurance and a certificate of road worthiness for the car. Obviously, two trust authorities are required:

- The insurance certificate is produced by an insurance company, say AXA.
- The certificate of road worthiness is produced by an accredited garage, say Joes Garage.

In reality, also the ownership of the car has to be recorded by the Driver and Vehicle Licensing Agency. However, this feature is suppressed here for ease of presentation.

The public/private key pairs for the two trust authorities are denoted by

$$(R_{\mathtt{AXA}}, s_{\mathtt{AXA}})\,, \qquad (R_{\mathtt{Joes}}, s_{\mathtt{Joes}})$$

with

$$R_{\mathtt{AXA}} = s_{\mathtt{AXA}}P, \qquad R_{\mathtt{Joes}} = s_{\mathtt{Joes}}P\,.$$

Suppose the owner of the car with registration number $\mathtt{X675}$ wished to obtain a new tax disk from the government. They could then log into some web site and claim that they had insured it through AXA and that Joes Garage had issued them with a certificate of road worthiness. The government could then email the user an encrypted version of the tax disc, upon payment of some fee, where the encryption is under the virtual trust authority $R_{\mathtt{AXA}} + R_{\mathtt{Joes}}$, and the identifier based public key is

$$Q_{\mathtt{X675}} = H_1(\mathtt{X675})\,.$$

Consequently, during the encryption of the tax disc m the ciphertext (U, V) is computed with $U = rP$ where r is a random element of $\mathbb{Z}_q$ and

$$V = m \oplus H_2\left(t\left(R_{\mathtt{AXA}} + R_{\mathtt{Joes}}, rQ_{\mathtt{X675}}\right)\right)\,.$$

The fact that the tax disc is encrypted enforces certain actions by the owner, who would need to obtain from each trust authority the corresponding (identifier based) private key

$$S_{\texttt{X675,AXA}} \text{ and } S_{\texttt{X675,Joes}} \text{ respectively}$$

with

$$S_{\texttt{X675,AXA}} = s_{\texttt{AXA}} Q_{\texttt{X675}}\,,$$

and

$$S_{\texttt{X675,Joes}} = s_{\texttt{Joes}} Q_{\texttt{X675}}\,.$$

Each trust authority delivers the private key only if the corresponding condition (insurance and road worthiness respectively) is satisfied. The owner now adds these private keys together to form a virtual private key, which allows the decryption of the electronic form of the tax disk by computing

$$\begin{array}{lll} V & \oplus & H_2\left(t\left(U, S_{\texttt{X675,AXA}} + S_{\texttt{X675,Joes}}\right)\right) = \\ V & \oplus & H_2\left(t\left(rP, \left(s_{\texttt{AXA}} + s_{\texttt{Joes}}\right) Q_{\texttt{X675}}\right)\right) = \\ V & \oplus & H_2\left(t\left(\left(s_{\texttt{AXA}} + s_{\texttt{Joes}}\right) P, rQ_{\texttt{X675}}\right)\right) = \\ V & \oplus & H_2\left(t\left(R_{\texttt{AXA}} + R_{\texttt{Joes}}, rQ_{\texttt{X675}}\right)\right) = m\,. \end{array}$$

Please note that in the example a traditional public key infrastructure is not necessary because the identifier based public key can be derived directly from the corresponding identifier.

3.2. Petri Net Representation

Workflows can advantageously be represented by Petri nets (see e. g. van der Aalst [1]). This subsection gives the Petri net for the example of Subsection 3.1. It thus illuminates the subprocess that is enforced by the use of identifier based cryptography. For the Petri net terminology the reader is referred to the book of van der Aalst and van Hee [4] and to van der Aalst [3]. Figures 1 and 2 present the Petri net for the example of Subsection 3.1.

The complete workflow net is composed of the nets PN_1 and PN_2. In the net PN_1 the government first produces the (unencrypted) document. This process is not specified in detail. The transition t^+ represents the task *government sends document to user*. The workflow net PN_1 alone can be viewed as an idealized situation where the user can be trusted totally and the government has no need to control the user. In the complete workflow the transition t^+ in PN_1 is refined by the workflow net PN_2 of Figure 2, i. e. t^+ is no longer a task but a reference to a subflow. The semantics of this hierarchical concept are straightforward; simply replace the refined transition t^+ by the corresponding subnet from the transition *government encrypts document* to the transition *user adds keys together and decrypts the encrypted document*. The subnet from the AND-split *user contacts AXA and Joes Garage* to the corresponding AND-join represents the cryptographic workflow. This part of the process is enforced by the use of identifier based cryptography. There are also two pairs of OR-splits/OR-joins. If the result of the checks are positive, the trust authorities AXA and Joes Garage send the correct (identifier based) private keys $S_{\texttt{X675,AXA}}$ and $S_{\texttt{X675,Joe}}$ respectively. Otherwise they send useless dummy keys. Only if the user has stored both correct private keys, the decryption in the final AND-join will produce the correct unencrypted document.

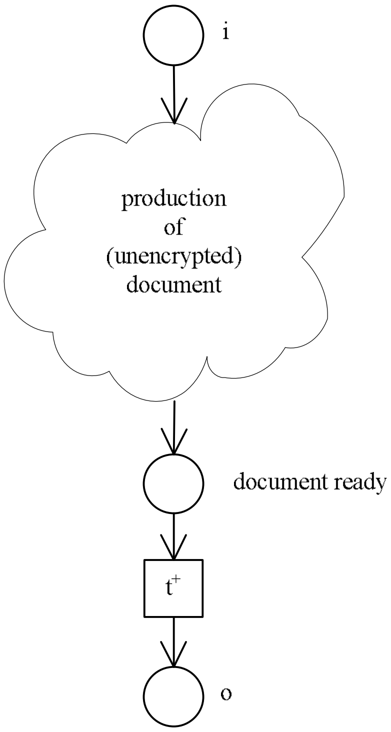

Figure 1. Workflow net PN_1 for the example of Subsection 3.1

Please note that the token has to carry a lot of information (for example the ciphertext). Hence, the nets PN_1 and PN_2 should be interpreted as colored Petri nets (see Jensen [14]).

3.3. Common Features of Cryptographic Workflows

The cryptographic workflow of Figure 2 is characterized by the fact that the user has to contact the two trust

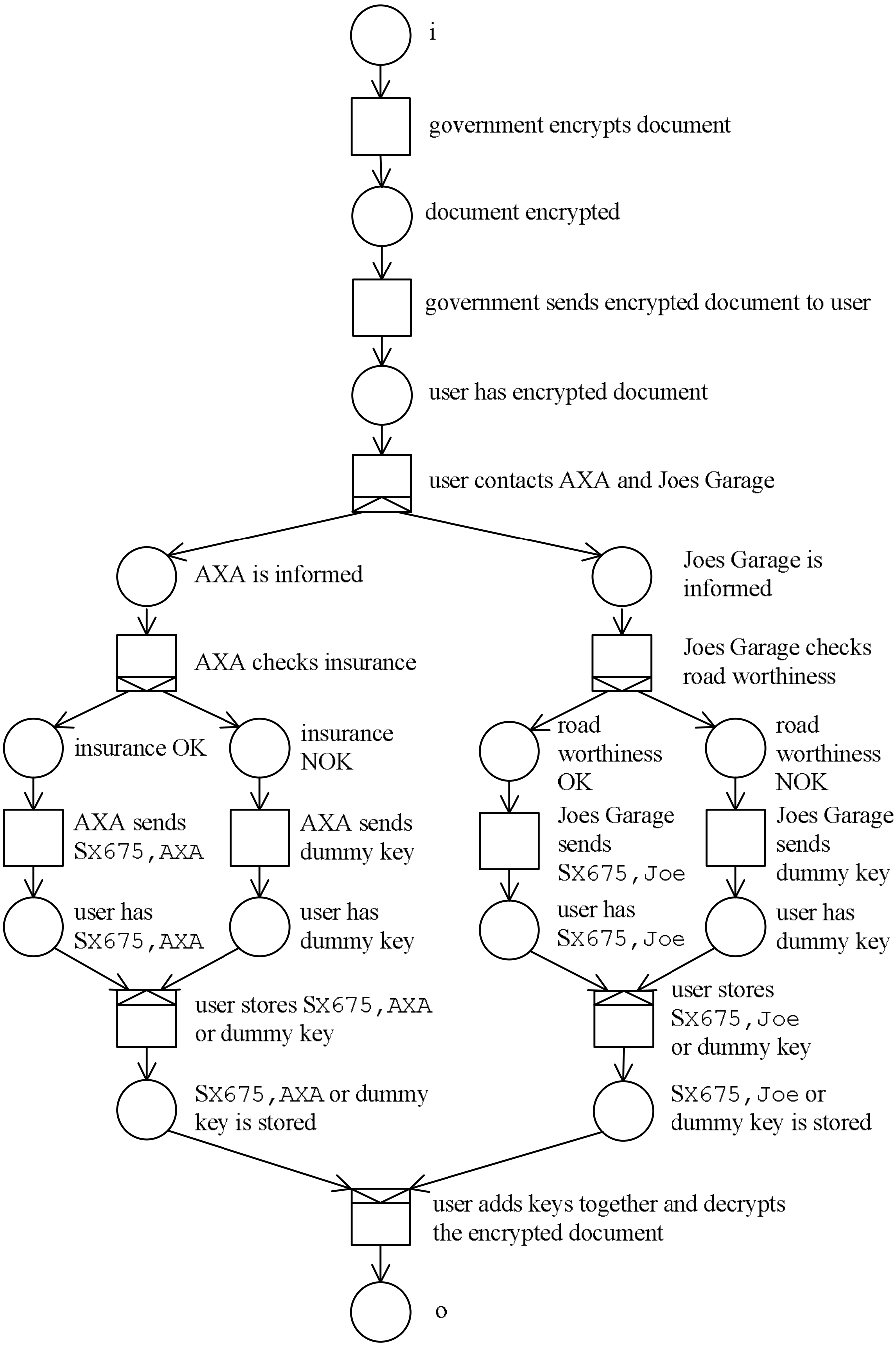
i
government encrypts document
document encrypted
government sends encrypted document to user
user has encrypted document
user contacts AXA and Joes Garage
AXA is informed
Joes Garage is informed
AXA checks insurance
Joes Garage checks road worthiness
insurance OK
insurance NOK
road worthiness OK
road worthiness NOK
AXA sends SX675,AXA
AXA sends dummy key
Joes Garage sends SX675,Joe
Joes Garage sends dummy key
user has SX675,AXA
user has dummy key
user has SX675,Joe
user has dummy key
user stores SX675,AXA or dummy key
user stores SX675,Joe or dummy key
SX675,AXA or dummy key is stored
SX675,Joe or dummy key is stored
user adds keys together and decrypts the encrypted document
o

Figure 2. Workflow net PN_2 for the example of Subsection 3.1

authorities AXA and Joes Garage to get the private keys

$$S_{\texttt{X675,AXA}} \quad \text{and} \quad S_{\texttt{X675,Joes}} .$$

Consequently, the enforced policy of this cryptographic workflow is defined by a logical conjunction. This is reflected in the AND-split/AND-join pair in Figure 2. In this example, the identifier X675 is the same for both keys. In a more general setting, there are n (possibly) different trust authorities $\texttt{TA}_i$ $(i = 1, 2, \ldots, n)$ with public/private key pairs $(R_{\texttt{TA}_i}, s_{\texttt{TA}_i})$ with

$$R_{\texttt{TA}_i} = s_{\texttt{TA}_i} P .$$

There are also n (possibly) different identifiers $\texttt{ID}_i$ with respect to the corresponding trust authorities $\texttt{TA}_i$. So, the identifier based key pairs $(Q_{\texttt{ID}_i}, S_{\texttt{ID}_i,\texttt{TA}_i})$ are linked via

$$S_{\texttt{ID}_i,\texttt{TA}_i} = s_{\texttt{TA}_i} Q_{\texttt{ID}_i} \quad \text{and} \quad Q_{\texttt{ID}_i} = H_1(\texttt{ID}_i) .$$

In this case, the message m is encrypted by the computations $U = rP$ and

$$V = m \oplus H_2 \left(\prod_{i=1}^{n} t(R_{\texttt{TA}_i}, rQ_{\texttt{ID}_i}) \right) .$$

Decryption is performed by computing

$$m = V \oplus H_2 \left(t(U, \sum_{i=1}^{n} S_{\texttt{ID}_i,\texttt{TA}_i}) \right) .$$

These equations generalize the corresponding relations in Subsection 2.2. For details the reader is referred to Chen et al. [10], Paterson [16] and [17]. The enforced policy is defined by a logical conjunction $\bigwedge_{i=1}^{n}$.

This approach is more efficient than the trivial technique that uses the following onion form of encryption where each encryption is applied in turn to obtain the ciphertext. Let us assume that the producer of the (unencrypted) document m wants to force the addressee to demonstrate the knowledge of the identifier based private keys $Y_1, Y_2, \ldots, Y_n$. Let y_i denote the proposition "recipient knows Y_i". Then, the policy enforcement can be described by the boolean formula

$$y_1 \wedge y_2 \wedge \ldots \wedge y_n \quad \text{(conjunction)} .$$

If $X_1, X_2, \ldots, X_n$ are the corresponding public keys, the producer would send

$$E_{X_1} \left(E_{X_2} \left(\ldots E_{X_n}(m) \ldots \right) \right)$$

to the addressee. Here, $E_K(m)$ denotes the encryption of m under the key K. Similarly, the policy

$$y_1 \vee y_2 \vee \ldots \vee y_n \quad \text{(disjunction)}$$

can be enforced by sending the set

$$\{E_{X_1}(m), E_{X_2}(m), \ldots, E_{X_n}(m)\} .$$

Hence, in both cases n applications of the encryption operation are necessary. In the case of disjunction a large increase in message size occurs. Also more general boolean functions consisting of conjunctions and disjunctions are possible. Let, for example, the policy enforcement be described by the logical formula

$$x \vee (y \wedge z) . \tag{1}$$

In this case the producer would send

$$\{E_X(m), E_Y(E_Z(m))\} ,$$

where X, Y and Z are the corresponding public keys. Clearly, one can never show not to know a given key. Hence, the logical formulae used can never contain the logical negation. For the most general case (combination of conjunctions and disjunctions), Smart [20] has presented solutions within the frame of identifier based encryption that are much more efficient than the trivial techniques described above, in particular with respect to bandwidth.

Key escrow is an inherent property of identifier based encryption as pointed out in Subsection 2.2. For this reason Al-Riyami and Paterson [6] present a protocol, which modifies the scheme of Boneh and Franklin [9] and avoids the problem of key escrow. This method, called *certificateless public key cryptography*, requires each user to have a (possibly unauthenticated) public key. Messages are then encrypted using a combination of a user's public key and its identifier based key. The system of Al-Riyami, Malone-Lee and Smart [5] uses a similar idea and employs secret sharing based monotone access structures such as that by Shamir [18] (in the case of threshold schemes) or Benaloh and Leichter [8] (in the case of general access structures). A monotone access structures can always be described by the appropriate disjunctive normal form boolean formula (see e. g. Stinson [21]). So, again the policy enforcements are defined by boolean formulas consisting of conjunctions and disjunctions. An analogous comment applies to the work of Barbosa and Farshim [7].

The above investigations show that all cryptographic workflows in current literature realize a policy enforcement that can be described by an underlying boolean formula, which is a combination of conjunctions and disjunctions. Consequently, in the most general case the Petri net descriptions of these workflows are built of a sequence construct and of pairs of AND-splits/AND-joins and OR-splits/OR-joins respectively. In the general case, the AND-split/AND-join pair in Figure 2 (simple conjunction) must be substituted by nested (one enclosed in the other) or disjoint pairs of AND-splits/AND-joins and OR-splits/OR-joins respectively. A typical situation is depicted in Figure 3. It corresponds to the logical formula of Equation (1). These observations motivate the following definition.

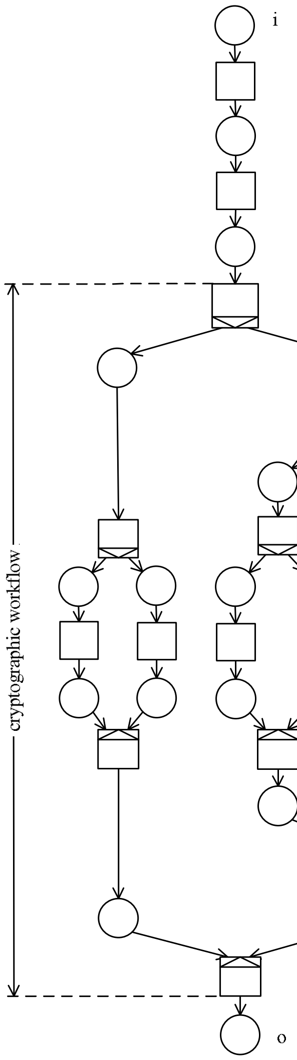

Figure 3. A typical cryptographic workflow net, cf. Equation (1)

Definition 1 A **cryptographic workflow net** is a workflow net that consists of (in this order)

a) a source place i,

b) a sequence construct,

c) nested (one enclosed in the other) or disjoint pairs of AND-splits/AND-joins and OR-splits/OR-joins respectively,

d) a sink place o.

Workflow nets are special cases of classic Petri nets (see e. g. van der Aalst and van Hee [4]). Part c) of the definition represents the **cryptographic workflow** itself, i. e. that part of the process that is enforced by the policy via identifier based cryptography.

4. Inheritance

Cryptographic workflows nets have many desirable properties that are given in the next theorem. Please note that for the application of the Petri net terminology the OR-splits and OR-joins have to be substituted by their explicit or implicit Petri net representations (cf. [4]). For the concept of free choice Petri nets the reader is referred to the book of Desel and Esparza [12]. The pair (PN_{cw}, i) denotes the system that consists of the Petri net PN_1 and the initial marking i.

Theorem 1 Let PN_{cw} be a cryptographic workflow net according to Definition 1. Then it is a) free choice, b) well-structured, c) S-coverable; d) (PN_{cw}, i) is safe; e) PN_{cw} is sound.

Proof. Throughout the proof, $\underline{PN_{cw}}$ denotes the short-circuited net of PN_{cw}.

a) If two transitions of PN_{cw} have a common input place, then both transitions have no other input places.

b) The short-circuited net $\underline{PN_{cw}}$ is well-handled. Indeed, let x and y be a pair of nodes of this net such that one is a place and the other a transition. Then by definition there cannot exist two disjoint elementary paths leading from x to y in $\underline{PN_{cw}}$.

c) For any node of the short-circuited net $\underline{PN_{cw}}$ there exists an S-component, which contains this node.

d) The net PN_{cw} with the initial marking i contains one token in the place i. Clearly, for each place the number of tokens never exceeds one.

e) Because of d) and Theorem 11 of van der Aalst [2] it is sufficient to demonstrate the liveness of $(\underline{PN_{cw}}, i)$: Indeed, for every reachable marking M' and every transition t there is a marking M'' reachable from M' that enables t. □

The complete workflow net is composed of the workflow net PN_1 of Figure 1 and a cryptographic workflow net (e. g. that of Figure 2 or Figure 3). Here, the task t^+ in PN_1 is refined by the cryptographic workflow net, i. e. the transition t^+ is replaced by the corresponding subnet. The precise semantics of this hierarchy concept are given in the following definition (cf. van der Aalst [3]).

Definition 2 (Compositionality) Let $PN_1 = (P_1, T_1, F_1)$ be the workflow net of Figure 1 and let $PN_2 = (P_2, T_2, F_2)$ be a cryptographic workflow net with $T_1 \cap T_2 = \emptyset$ and $P_1 \cap P_2 = \{i, o\}$. The composed workflow net $PN_3 = (P_3, T_3, F_3)$ is obtained by replacing transition t^+ in PN_1 by PN_2, i. e. $P_3 = P_1 \cup P_2$, $T_3 = (T_1 \setminus \{t^+\}) \cup T_2$ and

$$
\begin{aligned}
F_3 = {} & \\
& \big\{(x,y) \in F_1 \mid x \neq t^+ \wedge y \neq t^+\big\} \cup \\
& \{(x,y) \in F_2 \mid \{x,y\} \cap \{i,o\} = \emptyset\} \cup \\
& \big\{(x,y) \in P_1 \times T_2 \mid (x,t^+) \in F_1 \wedge (i,y) \in F_2\big\} \cup \\
& \big\{(x,y) \in T_2 \times P_1 \mid (t^+,y) \in F_1 \wedge (x,o) \in F_2\big\} .
\end{aligned}
$$

In the situation characterized by this definition the question arises how properties of the workflow net PN_1 are inherited by the complete workflow net PN_3 . Theorem 3 of [3] gives partial answers to this question. However, the special structure of the workflow nets in the present paper allows stronger results in the following theorem. Please note that the properties of the workflow net PN_1 of Figure 1 are determined mainly by the properties of the subprocess *production of (unencrypted) document*.

Theorem 2 (Inheritance) Let $PN_1 = (P_1, T_1, F_1)$ be the workflow net of Figure 1 and let $PN_2 = (P_2, T_2, F_2)$ be a cryptographic workflow net. Further, let $PN_3 = (P_3, T_3, F_3)$ be the composed workflow net according to Definition 2. Then the following statements hold:

a) If PN_1 is free choice, then PN_3 is free choice.

b) If PN_1 is well-structured, then PN_3 is well-structured.

c) If PN_1 is S-coverable, then PN_3 is S-coverable.

d) If (PN_1, i) is safe, then (PN_3, i) is safe.

e) If PN_1 is sound, then PN_3 is sound.

Proof.

a) The net PN_1 is free choice by assumption; PN_2 is free choice because of Theorem 1. Further, there does not exist a pair of transitions $(t_1, t_2) \in T_3 \times T_3$ with a common input place and with one transitioin in $T_3 \cap (T_1 \setminus \{t^+\})$ and the other in $T_3 \cap T_2$. Consequently, PN_3 is free choice.

b) Let x and y be a pair of nodes of PN_3 such that one is a place and the other a transition. Because PN_1 and PN_2 are both well-structured (for the latter cf. Theorem 1), there cannot exist two disjoint elementary paths leading from x to y in the short-circuited net $\underline{PN_3}$, if x and y are both in $(P_3 \cup T_3) \cap (P_1 \cup (T_1 \setminus \{t^+\}))$ or both in $(P_3 \cup T_3) \cap (P_2 \cup T_2)$. So, let x be in $(P_3 \cup T_3) \cap (P_1 \cup (T_1 \setminus \{t^+\}))$ and y be in $(P_3 \cup T_3) \cap (P_2 \cup T_2)$. Two paths from x to y must then have the node *document ready* in common. Similarly, if x is in $(P_3 \cup T_3) \cap (P_2 \cup T_2)$ and y is in $(P_3 \cup T_3) \cap (P_1 \cup (T_1 \setminus \{t^+\}))$, two paths from x to y must then have the short-circuit in common. So, in both cases the two paths are not disjoint and PN_3 is proven to be well-structured.

c) Each S-component of the short-circuited net $\underline{PN_1}$ can be combined with an S-component of $\underline{PN_2}$ (cf. Theorem 1) to an S-component of $\underline{PN_3}$.

d) Because (PN_1, i) is safe, the number of tokens of (PN_3, i) does not exceed 1 for each place in $P_3 \cap P_1$. In particular, this is valid for the place *document ready*. Because of Theorem 1, (PN_2, i) is safe, too. Consequently, for each place in $P_3 \cap P_2$ the number of tokens does not exceed 1, either.

e) The soundness of PN_1 guarantees that for all markings reachable from state i the place o will never contain more than one token. Consequently, the place *document ready* must be safe. This is the only input place of t^+. The rest follows from the soundness of PN_2 (Theorem 1) and from the proof of 3. of Theorem 3 in [3]. □

5. Conclusion

The examples in the introduction make clear that policy enforcement via identifier based cryptography (or its variants) is an attractive tool for electronic government. The corresponding processes can be modeled concisely by Petri nets. The present paper shows that the Petri nets associated with policy enforcements (called cryptographic workflow nets) have attractive features. The most important among these properties is soundness. Roughly speaking, a process, which is modeled by a sound workflow net, fulfills three requirements: (1) For every state reachable from the source place i it is possible - by performing a number of tasks - to come to a state in which there is a token in the sink place o; (2) when there is a token in this sink place o, all other tokens have disappeared; (3) there are no dead tasks, i. e. it is possible to execute an arbitrary task by following the appropriate route through the workflow net. These properties ensure that every case that begins at the place i will eventually be completed properly. Soundness is a strong indication that the workflow net is correctly established.

Further, the present paper demonstrates important inheritance properties: Let a government enforce certain actions of a citizen via the techniques of identity based cryptography. Then, on the modeling level so called policy enforcements are incorporated into an original workflow net. The results of Section 4 show that the extended net receives all the standard attractive features, that are present in the original net. In particular the extended net will have the free choice and the well structuredness properties, if these are valid for the original net. Consequently, in these nets it can be decided in polynomial time whether they are sound or not. If the original net is already known to be sound, the same can be guaranteed for the extended net. In other words, the employment of cryptographic workflows preserves desirable properties. So, from a modeling perspective the incorporation of policy enforcements does not produce any complications. This attractive feature stimulates the application of the suggested techniques and promotes the use of Petri nets in the practice of electronic government.

As a next step in future work the techniques of identity based cryptography have to be made accessible in Petri net based commercial workflow management systems like COSA [11]. On the academic level this has been done in the work of Wünschmann [23] for *CPN Tools* [15], a computer tool for constructing and analyzing colored Petri nets. Here Wünschmann makes use of the CPN ML library *Comms/CPN*, which enables *CPN Tools* to communicate with external applications and processes on the basis of the TCP/IP protocol.

6. Acknowledgement

The author is supported by the European Regional Development Fund - Europäischer Fonds für regionale Entwicklung (EFRE).

References

[1] W. M. P. van der Aalst. Three good reasons for using a Petri-net-based workflow management system. In: S. Navathe and T. Wakayama (Eds.), Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96), 179–201, Cambridge, Massachusetts, 1996.

[2] W. M. P. van der Aalst. Verification of workflow nets. In: P. Azéma and G. Balbo (Eds.), Application and Theory of Petri Nets 1997, Lecture Notes in Computer Science 1248, 407–426, Springer-Verlag, Berlin, 1997.

[3] W. M. P. van der Aalst. Workflow verification: Finding control-flow errors using Petri-net-based techniques. In: W. M. P. van der Aalst, J. Desel, and A. Oberweis (Eds.), Business Process Management: Models, Techniques, and Empirical Studies, Lecture Notes in Computer Science 1806, 161–183, Springer-Verlag, Berlin, 2000.

[4] W. M. P. van der Aalst and K. van Hee. Workflow Management. MIT-Press, Cambridge, Massachusetts, 2004.

[5] S. S. Al-Riyami, J. Malone-Lee, and N. P. Smart. Escrow-free encryption supporting cryptographic workflow. International Journal of Information Security, 5(4): 217–229, Springer-Verlag, Berlin, 2006.

[6] S. S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In: C. S. Laih (Ed.), Advances in Cryptology – ASIACRYPT 2003, Lecture Notes in Computer Science 2894, 452–473, Springer-Verlag, Berlin, 2003.

[7] M. Barbosa and P. Farshim. Secure cryptographic workflow in the standard model. In: R. Barua and T. Lange (Eds.), INDOCRYPT 2006, Lecture Notes in Computer Science 4329, 379–393, Springer-Verlag, Berlin, 2006.

[8] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In: S. Goldwasser (Ed.), Advances in Cryptology – CRYPTO 1988, Lecture Notes in Computer Science 403, 27–35, Springer-Verlag, Berlin, 1990.

[9] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In: J. Kilian (Ed.), Advances in Cryptology – CRYPTO 2001, Lecture Notes in Computer Science 2139, 213–229, Springer-Verlag, Berlin, 2001.

[10] L. Chen, K. Harrison, D. Soldera, and N. P. Smart. Application of multiple trust authorities in pairing based cryptosystems. In: G. Davida, Y. Frankel, and O. Rees (Eds.), Infrastructure Security: InfraSec 2002, Lecture Notes in Computer Science 2437, 260–275, Springer-Verlag, Berlin, 2002.

[11] COSA BPM Suite Roadmap. BPS-Solutions GmbH, Pulheim, Germany, 2009.

[12] J. Desel and J. Esparza. Free Choice Petri Nets. Cambridge University Press, Cambridge UK, 2005.

[13] G. Frey, M. Müller, and H. Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. IEEE Transactions on Information Theory, 45(5), 1717–1719, 1999.

[14] K. Jensen. Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use. 3 Volumes. Springer-Verlag, Berlin, 1997.

[15] K. Jensen, L. M. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. International Journal on Software Tools for Technology Transfer (STTT), 9(3-4), 213–254, 2007.

[16] K. G. Paterson. Cryptography from Pairings: A Snapshot of Current Research. Information Security Technical Report 7, 41–54, 2002.

[17] K. G. Paterson. Cryptography from pairings. In: I. F. Blake, G Seroussi, and N. P. Smart (Eds.), Advances in Elliptic Curve Cryptography, London Mathematical Soc. Lecture Note Series, Vol. 317, Cambridge University Press, 215–251, 2005.

[18] A. Shamir. How to share a secret. Communications of the ACM, 22(11), 612–613, 1979.

[19] A. Shamir. Identity based cryptosystems and signature schemes. In: G. R. Blakley and D. Chaum (Eds.), Advances in Cryptography – CRYPTO 1984, Lecture Notes in Computer Science 196, 47–53, Springer-Verlag, Berlin, 1985.

[20] N. P. Smart. Access control using pairing based cryptography. In: M. Joye (Ed.), Topics in Cryptology – CT-RSA 2003, Lecture Notes in Computer Science 2612, 111–121, Springer, Berlin, 2003.

[21] D. R. Stinson. Cryptography – Theory and Practice. Chapman & Hall/CRC, Boca Raton, FL, 2006.

[22] H. M. W. Verbeek, T. Basten, and W. M. P. van der Aalst. Diagnosing workflow processes using Woflan. The Computer Journal 44(4), 246–279, 2001.

[23] S. Wünschmann. Identitätsbasierte Kryptographie für sicheres Workflow-Management. Diploma thesis, University of Regensburg, 2007.