Mehrparteienberechnungen und Tschebyscheff-Polynome

Peter Lory

Institut für Wirtschaftsinformatik,
Universität Regensburg, D-93040 Regensburg, Deutschland

Abstract: The paper studies the potential of Chebyshev polynomials in the context of *privacy preserving data mining*: First the secure computation of the mean in a two-party scenario, second the secure distributed learning of a decision tree in both the two-party and the multi-party case. The investigations demonstrate that considerable gains in efficiency can be achieved in the design of the protocols owing to the better approximation quality of the truncated Chebyshev series in comparison to the truncated Taylor series.

1 Einleitung

„Chebyshev polynomials are everywhere dense in numerical analysis.“

Dieser leicht ironischen Bemerkung, deren Ursprung nicht genau zurück verfolgt werden kann, wird sicher kaum ein Kundiger widersprechen. In der Tat spielen die 1853 von Pafnuti Lwowitsch Tschebyscheff (1821–1894) vor der Akademie von St. Petersburg zum ersten Mal vorgestellten Polynome in der Numerischen Mathematik eine zentrale Rolle. Nach dem Wissen des Autors wurden sie jedoch bisher in der Kryptographie nicht verwendet. Die vorliegende Abhandlung legt dar, wo sie im Bereich des sicheren Mehrparteienberechnungen mit Gewinn eingesetzt werden können.

Sicheren Mehrparteienberechnungen liegt ein Szenario zu Grunde, bei dem eine Anzahl von verschiedenen, aber durch vertrauliche und authentische Kanäle miteinander verbundene Parteien versuchen, eine Funktion gemeinsam zu berechnen. Dabei soll von den Daten der einzelnen Parteien nicht mehr bekannt werden als aus dem gemeinsam berechneten Funktionswert ohnehin geschlossen werden kann. Wir stellen uns also vor, die Auswertung eines Protokolls käme unter die Attacke eines Angreifers, dem es gelingt eine oder mehrere Parteien unter seine Kontrolle zu bringen. Ein *passiver* Angreifer, im Englischen auch *semi-honest* oder *honest but curious* genannt, folgt dabei den Instruktionen des Protokolls getreu, versucht aber, aus den Informationen, die er während der Ausführung des Protokolls gewinnt, Schlüsse auf den Input einer anderen Partei zu ziehen. Mit Hilfe von Standard-Techniken können Protokolle, welche sicher gegen einen passiven Angreifer sind, unter gewissen Einschränkungen auch sicher gemacht werden gegen einen aktiven (*malicious*) Angreifer, der auch von den Instruktionen des Protokolls abweicht. Bezüglich einer formalen und ausführlichen Behandlung dieser Fragestellungen sei auf das Buch von Goldreich (2004) verwiesen.

Der vorliegenden Beitrag untersucht Protokolle aus dem Bereich des *privacy preserving Data-Mining*, deren Effizienz durch den Einsatz von Tschebyscheff-Polynomen verbessert werden kann. Dies hat potentielle Anwendungen im *Cloud-Computing*, bei dem aus Datenschutzgründen empfohlen wird, Datenbanken nicht bei einem einzelnen Dienstleister sondern in partitionierter Form zu speichern. [1] Den Protokollen, von denen im vorliegenden Beitrag ausgegangen wird, ist gemeinsam, dass aus dem Abbruch von Taylor-Reihen entstandene Polynome als Approximationen für andere Funktionen verwendet werden. Es erweist sich als wesentlich günstiger, nicht von den Taylor-Reihen, sondern von Entwicklungen nach Tschebyscheff-Polynomen auszugehen. Es ist dann bei gleicher Approximationsgüte möglich, deutlich früher abzubrechen. Dies resultiert in approximierenden Polynomen mit einem wesentlich niedrigeren Grad. Damit werden offensichtlich deutliche Effizienzvorteile erzeugt im Hinblick sowohl auf den Rechen- als auch auf den Kommunikationsaufwand der entstehenden Protokolle. Da bei diesen Modifikationen nur öffentlich bekannte Parameter verwendet werden, wird die Sicherheit der Ausgangsprotokolle nicht verschlechtert. Es sei darauf hingewiesen, dass im Bereich des *privacy preserving Data-Mining* die beteiligten Parteien ein Interesse an dem zu ermittelnden Ergebnis haben und daher erwartet werden kann, dass sie das Protokoll einhalten. Andererseits ist es für einen Angreifer, der in den Rechner

[1] Man vergleiche dazu etwa das Interview mit J. Müller-Quade in der Zeitschrift *Digital*, Ausgabe Juli/August 2011.

einer Partei eingedrungen ist, sehr schwierig, von einem spezifizierten Programm abzuweichen, das in einer komplexen Anwendung verborgen ist. In diesem Sinne ist in den vorliegenden Fällen das Szenario des *passiven* Angreifers ein realitätsnahes Modell. Darauf hat schon Pinkas (2003) hingewiesen.

In Abschnitt 2 und in Teilabschnitt 3.2 sind in einem Zwei-Parteien-Szenario Polynome der Gestalt

$$Z_d = \sum_{i=0}^{d} a_i \epsilon^i \tag{1}$$

sicher auszuwerten. Dabei wird *Oblivious Polynomial Evaluation* (Abk. OPE) nach Naor und Pinkas (1999 und 2006) verwendet. Dies ist ein Protokoll, das mit zwei Parteien abläuft: Ein Sender hat als Input ein Polynom P des Grades d über einem endlichen Körper $\mathcal{F}$; der Input des Empfängers ist ein Wert $\alpha \in \mathcal{F}$. Durch das Protokoll lernt der Empfänger $P(\alpha)$ (und nur dies), und der Sender lernt nichts. Dabei kommen Protokolle für *Oblivious Transfer* zum Einsatz (vgl. etwa Naor und Pinkas, 2005). Eine standardmäßige Anwendung von OPE ist die Erzeugung von additiven *Shares* des oben definierten Z_d, wobei γ_1 and γ_2 additive *Shares* von ϵ sind. Alle Koeffizienten und *Shares* sind Elemente des endlichen Körpers $\mathcal{F}$. Ein Protokoll, welches diese Aufgabe bewerkstelligt, ist gegeben in Abbildung 1 (siehe etwa Lindell und Pinkas, 2002).

1. P1 wählt zufällig b_1 und definiert das Polynom

$$P(X) = \sum_{i=0}^{d} a_i(\gamma_1 + X)^i - b_1 \,.$$

2. P1 führt mit P2 ein OPE-Protokoll durch. Spieler P2 lernt dabei $b_2 = P(\gamma_2)$, und P1 lernt nichts.

Am Ende haben die beiden Parteien additive *Shares* b_1 und b_2 von Z_d.

Abbildung 1: *Sharing* von Z_d aus (1), wobei $\epsilon = \gamma_1 + \gamma_2$.

Notation. Für eine Menge S (in Abschnitt 3 insbesondere eine Kollektion von Datensätzen) ist $|S|$ die Anzahl der Elemente von S. Der duale (natürliche) Logarithmus wird mit log (ln) bezeichnet. Das Symbol $T_\nu(x)$ steht für das Tschebyscheff-Polynom erster Art vom Grade ν mit $-1 \le x \le 1$.

Übersicht. Der vorliegende Beitrag untersucht das Potential der Tschebyscheff-Polynome zur Beschleunigung von Protokollen aus dem Bereich des *privacy preserving Data-Mining* anhand von zwei Beispielen: Abschnitt 2 behandelt ein Zwei-Parteien-Protokoll zur Berechnung des arithmetischen Mittels; Abschnitt 3 ist der sicheren verteilten Konstruktion von Klassifikationsbäumen sowohl im Fall von zwei als auch im Fall von mehreren Parteien gewidmet. Der Ausblick in Abschnitt 4 lenkt den Blick auf weitere Anwendungsmöglichkeiten.

2 Sichere verteilte Berechnung des arithmetischen Mittels

Kiltz, Leander und Malone-Lee (2005) haben ein Protokoll vorgestellt zur verteilten Berechnung des arithmetischen Mittels, welches sicher gegen einen passiven Angreifer ist. Dabei sind die Beiträge auf zwei Spieler verteilt: Spieler P1 hat n_1 Einträge in seiner Datenbank und P2 hat n_2. Diese Einträge seien bezeichnet mit $\{x_{1,1}, x_{1,2}, \ldots, x_{1,n_1}\}$ beziehungsweise mit $\{x_{2,1}, x_{2,2}, \ldots, x_{2,n_2}\}$. Die Summen x_1 und x_2 seien definiert als

$$x_1 = \sum_{i=1}^{n_1} x_{1,i} \quad \text{und} \quad x_2 = \sum_{i=1}^{n_2} x_{2,i}\,.$$

Ohne Einschränkung der Allgemeinheit kann angenommen werden, dass x_1 und x_2 ganzzahlig sind. Andernfalls ist geeignet zu erweitern. Die Aufgabe ist die Berechnung des Mittels

$$M = \frac{x_1 + x_2}{n_1 + n_2}, \tag{2}$$

wobei (x_1, n_1) dem Spieler P1 bekannt ist und (x_2, n_2) dem Spieler P2. Der entscheidende Punkt im Protokoll ist die verteilte Berechnung der in (2) auftretenden Division. Zur Berechnung einer beliebig genauen Approximation $\hat{M}$ von M geht das Protokoll dabei folgendermaßen vor: Es sei m_1 die n_1 nächstgelegene Zweierpotenz, d. h.

$$2^{m_1-1} + 2^{m_1-2} \le n_1 < 2^{m_1} + 2^{m_1-1}\,,$$

und m_2 sei analog definiert. Es sei $k = \max\{m_1, m_2\} + 1$. Mit der Definition

$$\epsilon := 1 - (n_1 + n_2)/2^k \tag{3}$$

haben wir

$$n_1 + n_2 = 2^k(1 - \epsilon)\,, \quad \text{wobei} \quad -\frac{1}{2} < \epsilon \le \frac{5}{8}\,. \tag{4}$$

Somit

$$\frac{2^k}{n_1 + n_2} = \frac{1}{1 - \epsilon} = \sum_{i=0}^{\infty} \epsilon^i = \sum_{i=0}^{d} \epsilon^i + R_d\,, \tag{5}$$

wobei

$$|R_d| \le \frac{5}{3}\left(\frac{5}{8}\right)^d\,. \tag{6}$$

Gleichung (5) wird dann mit einer hinreichend großen Zahl multipliziert, um die Berechnungen im Bereich der ganzen Zahlen zu halten. Die sichere verteilte Auswertung des dabei aus der abgebrochenen geometrischen Reihe $\sum_{i=0}^{d} \epsilon^i$ entstehenden Polynoms erfolgt durch das Protokoll von Abbildung 1. Bezüglich der Einzelheiten sei auf die Artikel von Kiltz, Leander und Malone-Lee (2005) und Naor und Pinkas (1999 und 2006) verwiesen.

Die Auswertung des erwähnten Polynoms kann beschleunigt werden, indem man nicht von der geometrischen Reihe, sondern von der entsprechenden Tschebyscheff-Entwicklung ausgeht. Durch die Substitution $\epsilon = (9x + 1)/16$ wird das Intervall

in (4) abgebildet auf das Standardintervall $-1 < x \leq 1$ und $1/(1-\epsilon)$ wird transformiert zu

$$f(x) := \frac{16}{15} \cdot \frac{1}{1 - \frac{3x}{5}} \,. \tag{7}$$

Diese Funktion kann analytisch fortgesetzt werden zur komplexen Funktion $f(z)$ mit $z = x+iy$ in der Ellipse mit den Brennpunkten $z = \pm 1$, der großen Halbachse $a = 5/3$ und der Exzentrizität $e = 1/a$. Durch die Transformation

$$z = \frac{1}{2}\left(\zeta + \frac{1}{\zeta}\right) \tag{8}$$

wird der Kreisring $1/\rho < |\zeta| < \rho$ mit $\rho := a + \sqrt{a^2 - 1} = 3$ in der ζ-Ebene abgebildet zur oben erwähnten (doppelt überdeckten) Ellipse in der z-Ebene und

$$f\left(\frac{1}{2}\left(\zeta + \frac{1}{\zeta}\right)\right) = \frac{16}{15} \cdot \frac{-10\zeta}{3\zeta^2 - 10\zeta + 3} = \frac{4}{3} \cdot \frac{1}{1 - \frac{\zeta}{3}} + \frac{4}{9} \cdot \frac{1}{\zeta} \cdot \frac{1}{1 - \frac{1}{3\zeta}}.$$

Dies liefert die Laurent-Entwicklung

$$f\left(\frac{1}{2}\left(\zeta + \frac{1}{\zeta}\right)\right) = \frac{8}{3}\left[\frac{1}{2}\sum_{\nu=0}^{\infty}\left(\frac{1}{3}\right)^{\nu}\zeta^{\nu} + \frac{1}{6}\sum_{\nu=1}^{\infty}\left(\frac{1}{3}\right)^{\nu-1}\frac{1}{\zeta^{\nu}}\right]$$

in dem oben erwähnten Kreisring. Daraus folgt die Tschebyscheff-Entwicklung

$$f(z) = \frac{8}{3}\left[\frac{1}{2} + \sum_{\nu=1}^{\infty}\left(\frac{1}{3}\right)^{\nu} T_{\nu}(z)\right]. \tag{9}$$

(siehe z. B. Bulirsch und Stoer (1968) oder Mason und Handscomb (2003)). Daraus erhalten wir

$$\frac{2^k}{n_1 + n_2} = \frac{1}{1-\epsilon} = \frac{8}{3}\left[\frac{1}{2} + \sum_{\nu=1}^{\hat{d}}\left(\frac{1}{3}\right)^{\nu} T_{\nu}(x)\right] + \hat{R}_{\hat{d}} \tag{10}$$

mit

$$|\hat{R}_{\hat{d}}| \leq \frac{4}{3}\left(\frac{1}{3}\right)^{\hat{d}}. \tag{11}$$

Es ist unmittelbar zu erkennen, dass das Restglied $\hat{R}_{\hat{d}}$ von (11) wesentlich schneller gegen 0 strebt als das Restglied R_d in (6). Gleichsetzen der oberen Schranken für diese beiden Restglieder liefert $\hat{d} \approx 0.43 \cdot d - 0.2$. Eine weitere Vernachlässigung des kleinen additiven Terms führt zu

$$\hat{d} \approx 0.43 \cdot d \,. \tag{12}$$

Dies zeigt, dass durch die Verwendung der Tschebyscheff-Entwicklung bei gleicher Genauigkeitsanforderung der Grad $\hat{d}$ deutlich kleiner gewählt werden kann als d. Dies bewirkt im Protokoll deutliche Einsparungen sowohl im Rechen- als auch im Kommunikationsaufwand. Dazu muss vorher noch die abgebrochene Tschebyscheff-Entwicklung in (10) in ein Polynom vom Grad $\hat{d}$ umgewandelt werden, dessen Koeffizienten noch durch die Multiplikation mit einer geeigneten Zahl ganzzahlig zu machen sind. Die sichere verteilte Auswertung erfolgt wieder durch das Protokoll von Abbildung 1 in einem hinreichen großen endlichen Körper $\mathcal{F}$.

3 Sichere verteilte Klassifikation

3.1 Grundlagen der Induktion von Entscheidungsbäumen

Die Induktion von Entscheidungsbäumen erfolgt in der Regel rekursiv im Top-Down-Prinzip. Bei jedem Schritt wird dabei das Attribut gesucht, mit welchem sich die Trainingsdaten in diesem Schritt bezüglich des Zielattributs am besten klassifizieren lassen. Der von Quinlan (1986) eingeführte grundlegende Algorithmus *ID3* und sein Nachfolger *C4.5* verwenden als Maß für die Bestimmung der *besten* Klassifizierung ein Entropie-Maß, den Informationsgewinn *Gain(S,A)* für das Attribut A bei der aktuellen Kollektion S von Trainingsdatensätzen. Dieser ist definiert durch

$$Gain(S, A) := Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)\,. \tag{13}$$

Dabei ist *Values(A)* die Menge aller möglichen Werte (Ausprägungen) für das Attribut A, und S_v ist diejenige Teilmenge von S, für die das Attribut A den Wert v hat. Die Entropie einer Kollektion S von Datensätzen relativ zu einer Klassifikation mit den Klassen $c_1, c_2, \ldots, c_l$ ist definiert durch

$$Entropy(S) := \sum_{i=1}^{l} -\frac{|S(c_i)|}{|S|} \log \frac{|S(c_i)|}{|S|}\,, \tag{14}$$

wobei $S(c_i)$ die Teilmenge der zur Klasse c_i gehörenden Datensätze ist. Bezüglich der Einzelheiten sei der Leser auf das Buch von Mitchell (1997) verwiesen. Nach Wu et alii (2008) gehört *C4.5* zu den Top-10-Algorithmen im *Data-Mining*.

3.2 Der Zwei-Parteien-Fall

Wenn wir uns die Datensätze einer Datenbank als Zeilen vorstellen, so spricht man von einer **horizontalen Partitionierung**, wenn ein Teil dieser Zeilen bei der Partei P1 und der andere bei der Partei P2 gespeichert ist. Für diesen Fall haben Lindell und Pinkas (2002, 2009) ein Protokoll vorgestellt, das es den beiden Parteien erlaubt, den Klassifikationsbaum sicher und verteilt mit Hilfe des *ID3*-Algorithmus zu ermitteln, ohne dass - im Sinne eines passiven Angreifers - die eine Partei mehr über die Daten der anderen Partei erfährt als durch den gemeinsamen Output (den Klassifikationsbaum) enthüllt wird. Der wesentliche Schritt dabei ist der Vergleich der Größen $Gain(S, A)$ für verschiedene Attribute A. Da es dabei nur um den Vergleich von Werten geht und $Entropy(S)$ nicht von dem Attribut abhängt, ist nur der zweite Summand auf der rechten Seite von (13) relevant. Dieser zweite Summand lautet

$$\sum_{v \in Values(A)} \frac{|S_v|}{|S|} \sum_{i=1}^{l} -\frac{|S_v(c_i)|}{|S_v|} \log \frac{|S_v(c_i)|}{|S_v|} \tag{15}$$

und kann umgeformt werden zu

$$\frac{1}{|S|} \left(- \sum_{v \in Values(A)} \sum_{i=1}^{l} |S_v(c_i)| \log |S_v(c_i)| \quad + \sum_{v \in Values(A)} |S_v| \log |S_v| \right). \tag{16}$$

Da es nur um einen Vergleich geht, kann dies mit mit $|S| \ln 2$ multipliziert werden. Dadurch entsteht

$$- \sum_{v \in Values(A)} \sum_{i=1}^{l} |S_v(c_i)| \ln |S_v(c_i)| \quad + \sum_{v \in Values(A)} |S_v| \ln |S_v| \,. \tag{17}$$

Von der Menge S_v der Datensätze mit dem Attributwert v für das Attribut A gehöre die Menge $S_v^{(1)}$ zur Datenbank des Spielers P1 und $S_v^{(2)}$ zu der des Spielers P2. Es gilt also $|S_v| = |S_v^{(1)}| + |S_v^{(2)}|$ und ebenso $|S_v(c_i)| = |S_v^{(1)}(c_i)| + |S_v^{(2)}(c_i)|$. Es ist also wiederholt die verteilte Auswertung von Termen der Form $x \ln x$ mit $x = u_1 + u_2$ nötig, wobei u_1 dem Spieler P1 bekannt ist und u_2 dem Spieler P2. Dazu gehen Naor und Pinkas (2002) folgendermaßen vor:

1. Durch das Protokoll von Yao (1986) auf der Basis von *Garbled Circuits* werden *Shares* für die x am nächsten gelegene Zweierpotenz n bestimmt und ebenfalls für ϵ, so dass $x = 2^n(1 + \epsilon)$. Es ist

$$-\frac{1}{4} \leq \epsilon < \frac{1}{2}. \tag{18}$$

2. Zur Approximation von $\ln(1+\epsilon)$ kommt eine Partialsumme der Taylor-Reihe für den natürlichen Logarithmus

$$\ln(1 + \epsilon) = \sum_{j=1}^{\infty} \frac{(-1)^{j+1} \epsilon^j}{j} \tag{19}$$

   zum Einsatz. Die sichere verteilte Auswertung der Partialsumme nach einer geeigneten, die Berechnungen ganzzahlig haltenden Multiplikation erfolgt durch das Protokoll von Abbildung 1. Damit liegen auch *Shares* für eine Näherung von $\ln x$ vor.

3. Durch ein Multiplikationsprotokoll auf der Basis von OPE werden *Shares* für eine Näherung von $x \ln x = x(n \ln 2 + \ln(1 + \epsilon))$ bestimmt.

Die Rechnungen finden statt in einem hinreichend großen endlichen Körper $\mathcal{F}$. Lindell und Pinkas (2002) schätzen den Fehler, der durch den Abbruch der Taylor-Reihe (19) entsteht, folgendermaßen ab:

$$\left| \ln(1 + \epsilon) - \sum_{j=1}^{d} \frac{(-1)^{j+1} \epsilon^j}{j} \right| < \frac{|\epsilon|^{d+1}}{d+1} \cdot \frac{1}{1 - |\epsilon|} \leq \frac{1}{2^d (d+1)} \,. \tag{20}$$

Dabei ist (18) zu beachten.
Es ist nun wieder wesentlich günstiger, anstatt von der Taylor-Reihe (19) von der entsprechenden Tschebyscheff-Entwicklung auszugehen. Geht man analog vor wie in Abschnitt 2, so erhält man

$$\ln(1 + \epsilon) = \ln \frac{3(3 + \sqrt{8})}{16} + \sum_{\nu=1}^{\infty} \frac{(-1)^{\nu+1} \cdot 2}{\nu (3 + \sqrt{8})^{\nu}} \cdot T_{\nu}(y) \,, \tag{21}$$

wobei $y = (8\epsilon - 1)/3$ und $-1 \leq y < 1$. Da die Koeffizienten hier wesentlich rascher fallen als in der Taylor-Reihe (19), ergibt sich für die abgebrochene Tschebyscheff-Entwicklung eine wesentlich günstigere Abschätzung:

$$\ln(1+\epsilon) = \ln \frac{3(3+\sqrt{8})}{16} + \sum_{\nu=1}^{\hat{d}} \frac{(-1)^{\nu+1} \cdot 2}{\nu(3+\sqrt{8})^{\nu}} \cdot T_{\nu}(y) + \hat{R}_{\hat{d}} \tag{22}$$

mit

$$|\hat{R}_{\hat{d}}| < \frac{2}{(3+\sqrt{8})^{\hat{d}}(\hat{d}+1)(2+\sqrt{8})}\,. \tag{23}$$

Die abgebrochene Tschebyscheff-Entwicklung in (22) ist wieder in ein Polynom vom Grad $\hat{d}$ umzuwandeln, dessen Koeffizienten noch durch die Multiplikation mit einer geeigneten Zahl ganzzahlig gehalten werden müssen. Die sichere verteilte Auswertung erfolgt wieder durch das Protokoll in Abbildung 1 in einem hinreichen großen endlichen Körper $\mathcal{F}$. Gleichsetzen der oberen Schranken in (20) und (23) liefert

$$\hat{d} \approx 0.393 \cdot d\,. \tag{24}$$

Dies zeigt wieder, dass durch die Verwendung der Tschebyscheff-Entwicklung bei gleicher Genauigkeitsanforderung der Grad $\hat{d}$ deutlich kleiner gewählt werden kann als d mit der Folge deutlicher Einsparungen sowohl im Rechen- als auch im Kommunikationsaufwand des Protokolls.

Der Fall einer **vertikal partitionierten** Kollektion der Datensätze, bei dem die Spalten bei unterschiedlichen Parteien gespeichert sind, wurde von Du und Zhan (2002) behandelt. Das dort vorgeschlagene Protokoll zur verteilten Berechnung des Logarithmus ist zwar effizient, leidet jedoch unter einem Leck geheimer Informationen. Darauf haben schon Kiltz, Leander und Malone-Lee (2005) hingewiesen.

3.3 Der Mehrparteien-Fall

Auch für den Fall von mehr als zwei Parteien kann die Tschebyscheff-Entwicklung gewinnbringend eingesetzt werden. Es seien dazu die additiven *Shares* $|S_v^{(j)}|$ beziehungsweise $|S_v^{(j)}(c_i)|$ des j-ten Spielers umgewandelt in polynomiale *Shares* im Sinne des *Threshold*-Schemas von Shamir (1979). Die Approximation der Logarithmus-Funktion erfolgt wie in Teilabschnitt 3.2 beschrieben durch ein ganzzahliges Polynom, wobei gemäß (24) die Verwendung der Tschebyscheff-Entwicklung bei gleicher Genauigkeitsanforderung einen deutlich niedrigeren Grad $\hat{d}$ erlaubt als der Abbruch der Taylor-Reihe bei Grad d. Dieses Polynom wird dann nach dem Horner-Schema ausgewertet. Dabei können die verteilten Additionen einfach lokal durch jeden Spieler durchgeführt werden. Für die sichere verteilte Multiplikationen muss ein komplizierteres Protokoll zum Einsatz kommen. Es seien hier genannt das Protokoll von Gennaro, Rabin und Rabin (1998) mit den effizienteren Varianten von Lory (2007 und 2009). Der Rechenzeitvergleich von Koschuch et alii (2011) gibt Hinweise, welche Version in einem gegebenen Szenario am besten geeignet ist.

4 Ausblick

An den Beispielen der sicheren Zwei-Parteien-Berechnung des arithmetischen Mittels und der sicheren verteilten Induktion des Klassifikationsbaums sowohl in einem Zwei-Parteien-Szenario als auch bei mehr als zwei Beteiligten wurde gezeigt, dass die Verwendung einer entsprechenden Tschebyscheff-Entwicklung ein geeignetes Mittel ist, um die Protokolle effizienter zu gestalten. Der wesentliche Faktor dabei ist die bessere Approximationseigenschaft der abgebrochen Tschebyscheff-Entwicklung im Vergleich zur abgebrochenen Taylor-Entwicklung. Ein weiteres Anwendungsbeispiel ist geheimes Lernen in Neuronalen Netzen nach Chang und Lu (2001). Dabei muss die Aktivitätsfunktion $\phi(\epsilon) = a \tanh(b\epsilon)$ stückweise durch Polynome niedrigen Grades approximiert werden.

Danksagung

Der Autor wurde gefördert durch den *Europäischen Fond für regionale Entwicklung (EFRE).*

Literaturverzeichnis

R. Bulirsch und J. Stoer. Darstellung von Funktionen in Rechenautomaten. In: R. Sauer und I. Szabó (Hrsg.), *Mathematische Hilfsmittel des Ingenieurs*, Band 141 der Grundlehren der mathematischen Wissenschaften, pp. 352–446, Springer, Berlin, 1968.

W. Du und Z. Zhan. Bulding decision tree classifier on private data. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), Workshop on Privacy, Security and Data Mining, Maebashi City, Japan, 2002.

Y.-C. Chang und C.-J. Lu. Oblivious polynomial evaluation und oblivious neural learning, in: C. Boyd (Hrsg.), Advances in Cryptology – ASIACRYPT 2001, Lecture Notes in Computer Science 2248, pp. 369–384, Springer, Berlin, 2001.

R. Gennaro, M. O. Rabin und T. Rabin. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In: Proceedings of the 17th ACM Symposium on Principles of Distributed Computing (PODC'98), pp. 101–111, ACM Press, 1998.

O. Goldreich. *Foundations of Cryptography: Volume 2 - Basic Applications*. Cambridge University Press, 2004.

E. Kiltz, G. Leander und J. Malone-Lee. Secure computation of the mean and related statistics. In: J. Kilian (Hrsg.), Proceedings of the 2nd Theory of Cryptography Conference (TCC'2005), Lecture Notes in Computer Science 3378, pp. 283–302, Springer, Berlin, 2005.

M. Koschuch, M. Hudler, M. Krüger, P. Lory und J. Wenzl. Optimizing cryptographic theshold schemes for the use in wireless sensor networks. In: M. S. Obaidat, J. L. Sevillano und E. C. Ortega (Hrsg.), Proceedings of DCNET 2011 – International Conference on Data Communication Networking, Seville, Spain, pp. 75–78, 2011.

Y. Lindell und B. Pinkas. Privacy preserving data mining. Journal of Cryptology, 15: 177–206, 2002.

Y. Lindell und B. Pinkas. Secure multiparty computation for privacy-preserving data mining. Journal of Privacy and Confidentiality, 1(1): 59–98, 2009.

P. Lory. Reducing the complexity in the distributed multiplication protocol of two polynomially shared values. In: Proceedings of the 3rd IEEE International Symosium on Security in Networks and Distributed Systems (SSNDS'2007), Band 1 von AINA'2007, pp. 404–408, IEEE Computer Society, 2007.

P. Lory. Secure distributed multiplication of two polynomially shared values: Enhancing the efficiency of the protocol. In: R. Falk, W. Goudalo, E. Y. Chen, R. Savola und M. Popescu (Hrsg.), Proceedings to the Third International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2009), IEEE, pp. 286–291, 2009.

J. C. Mason und D. C. Handscomb. *Chebyshev Polynomials*. Chapman & Hall/CRC, Boca Raton, 2003.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

M. Naor und B. Pinkas. Oblivious transfer and polynomial evaluation. In: J. S. Vitter, L. Larmore und T. Leighton (Hrsg.), Proceedings of the 31st ACM Symposium on Theory of Computing (STOC'99), pp. 245–254, ACM Press, 1999.

M. Naor und B. Pinkas. Computationally secure oblivious transfer. Journal of Cryptology, 18: 1–35, 2005.

M. Naor und B. Pinkas. Oblivious polynomial evaluation. SIAM Journal on Computing, 35(5): 1254–1281, 2006.

B. Pinkas. Cryptographic techniques for privacy-preserving data mining. ACM SIGKDD Explorations Newsletter, 4(2): 12–19, 2003.

J. R. Quinlan. Introduction to decision trees. Machine Learning, 1(1): 81–106, 1986.

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

X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand und D. Steinberg. Top 10 algorithms in data mining. Knowl Inf Syst, 14:1–37, 2008.

A. C. Yao. How to generate and exchange secrets. Proceedings of the 27th Symposium on Foundations of Computer Science (FOCS), IEEE, pp. 162–167, 1986.