Homomorphic Tallying with the ElGamal Cryptosystem

Flavio Bolfing
Software and Systems
University of Applied Sciences Chur, Switzerland
www.hsr.ch/mse

Abstract

Diese Seminararbeit stellt das homomorphe Auszählen mittels der ElGamal-Verschlüsselung bei E-Voting Systemen vor. Der Fokus ist vor allem auf den mathematischen Teil des Systems gesetzt worden und soll dem Leser einen Einstieg in E-Voting mit homomorphen Eigenschaften geben.

1. Mathematische und kryptografische Grundlagen

1.1. Euler'sche φ-Funktion

Die Euler'sche φ-Funktion gibt die Anzahl der Elemente eines reduzierten Restsystems modulo m an: Beispiel: Modulo 8={1,3,5,7} $\phi(8) = 4$. Die Euler'sche φ-Funktion hat in der Kryptographie eine besondere Bedeutung aufgrund einiger ihrer Eigenschaften:

Eigenschaft 1: p=Primzahl $\rightarrow$ $\phi(p) = p - 1$

Eigenschaft 2: $n = p \cdot q$ (p,q sind Primzahlen) $\rightarrow$ $\phi(n) = (p-1) \cdot (q-1)$

Wie wir aus der Eigenschaft 2 erkennen können, ist die Bestimmung von φ(n) aus zwei Primzahlen relativ einfach, während die Umkehrfunktion, also die Faktorisation von n sehr aufwändig zum berechnen ist. Auf diesem Umstand baut die Sicherheit einiger Systeme in der heutigen Kryptographie auf (z.B die RSA-Verschlüsselung).

1.2. Modulare Inverse

Definition: Die modulare Inverse eines Integers a mod m ist ein Integer x, so dass

$a^{-1} \equiv x \bmod m$ oder $ax \equiv 1 \bmod m$ $(a \cdot a^{-1} \equiv 1 \bmod m)$

lösbar für ein bestimmtes x ist. Integer a hat dabei nur dann eine modulare Inverse x mod m, wenn der ggT(a,m)=1 ist. Man sagt auch a und m sind koprim (teilerfremd) zueinander.

Modulare Inverse haben mit ihren Eigenschaften einen wichtigen Wert in der Kryptographie. Ist Integer a koprim zu modulo m (also ggT(a,m)=1), dann gilt:

1) Es gibt Integers x und y, so dass $bx + my = 1$
2) $b^{-1} \equiv b^{\phi(m)-1} \bmod m$
3) $b^{\phi(m)} \equiv 1 \bmod m \rightarrow$ Euler-Fermat Theorem

1.3. Kleiner fermatscher Satz (Engl. little Fermat´s theorem)

Der Umstand, dass man nebst modularer Addition und Multiplikation auch modulare Exponentation braucht, ist mit dem kleinen Satz von Fermat gegeben:

Gegeben: Primzahl p

$a^p \equiv a \bmod p \rightarrow$ Falls a und p koprim sind, gilt: $a^{p-1} \equiv 1 \bmod p \rightarrow a^{\phi(m)} \equiv 1 \bmod p$

1.4. Innere Verknüpfung

Nimmt man 2 Elemente (x und y) aus der Menge der natürlichen Zahlen, dann bilden die 2 Elemente zusammen das Paar (x,y). Diesem Paar wird nun ein x-beliebiges Element z (aus der Menge der natürlichen Zahlen) zugeordnet: $f : (x, y) \rightarrow z \quad = f(x,y)$

In diesem Fall wird also eine Addition von 2 natürlichen Zahlen x und y ausgeführt. Das Ergebnis z ist ebenfalls eine natürliche Zahl. Dies nennt man eine innere Verknüpfung $z = x \circ y$

1.5. Monoid

Gegeben sei eine Menge M, auf welche eine Verknüpfung $\circ$ definiert ist. Menge $(M, \circ)$ heisst Monoid, falls folgende Bedingungen erfüllt sind:

B1: Innere Verknüpfung: $a \circ b \in M$

B2: Assoziativ Gesetz erfüllt ist: $(a \circ b) \circ c = a \circ (b \circ c)$

B3: Es ein neutrales Element e gibt: $a \circ e = e \circ a = a$

Beispiel: $(N_0, +)$ B1: $0 + 1 = 2 \in M$

B2: $(0 + 1) + 2 = 0 + (1 + 2)$

B3: $3 + 0 = 0 + 3 = 3$

1.6. Gruppe

Ein Monoid heisst Gruppe, falls zu den 3 Monoid-Bedingungen noch eine 4te hinzukommt:

B1: Innere Verknüpfung: $a \circ b \in M$

B2: Assoziativ Gesetz erfüllt ist: $(a \circ b) \circ c = a \circ (b \circ c)$

B3: Es ein neutrales Element e gibt: $a \circ e = e \circ a = a$

B4: Es ein inverses Element a^{-1} gibt: $a \circ a^{-1} = a^{-1} \circ a = e$ (neutrales Element)

1.10. Ring

Eine algebraische Struktur $(M, +, \cdot)$ ist ein Ring, falls:

B1: (M,+) = Gruppe

B2: Das kommutativ Gesetz erfüllt ist: $a \circ b = b \circ a$

B3: $(M, \cdot)$ = Monoid

B4: Das distributiv Gesetz erfüllt ist: $a \cdot (b+c) = ab + bc$ und $(a+b) \cdot c = ac + bc$

1.11. Körper

Eine algebraische Struktur ist ein Körper, falls:

B1: Die 4 Bedingungen eines Ringes müssen erfüllt sein

B2: Neut. Element: 2 untersch. Elemente (0,1) mit der Eigenschaft, dass für $a \in F$ gilt: $0+a=a$ und $1 \bullet a=a$

B3: Inverses Element: Für jedes $a \varepsilon F$ existiert ein Element $-a \varepsilon F$ (= negativ von a), so dass $-a+a=0$

B3: Inverses Element: Für jedes $a \varepsilon F$ existiert ein Element a^{-1} (= inverses von a), so dass $a \cdot a^{-1} = 1$

1.12. Endliche Körper (Engl.: finite fields)

Endliche Körper (oder Galois Feld (GF)) sind Körper, in welchen die Menge F eine endliche Menge verkörpert (Bsp.: Menge der komplexen Zahlen ist eine unendliche Menge). Als ein

einfaches Beispiel gilt: GF(2) mit den Elementen F={1,2}. Ein arithmetischer Modulus einer Primzahl p ist ebenfalls ein endlicher Körper (GF(p)). Dieser Körper wird auch $\mathbb{Z} / p\mathbb{Z}$ genannt.

1.13.Homomorphismus

Ein Homomorphismus ist in der Algebra eine strukturerhaltende Abbildung. Dabei müssen folgende Bedingungen erfüllt sein:

Als Beispiel werden 2 algebraische Strukturen (M1,$\circ$) und (M2, $\bullet$) verwendet. Nun wird mit einer Funktion f eine Abbildung M1 $\rightarrow$ M2 gemacht. Diese Abbildung gilt dann als homomoprh, wenn gilt:

B1: $f(a) \cdot f(b)=f(a \circ b)$

B2: $f(e1) = e2 \rightarrow$ e1=Neutrales Element von M1 $\rightarrow$ e2=Neutrales Element von M1

Beispiel einer homomorphen Abbildung: ($\mathbb{Z}$,+) und ($\mathbb{R}$, $\bullet$)

Funktion $f = \mathbb{Z} \rightarrow \mathbb{R}$ $\quad f = x \rightarrow y$ wobei z.B $f(x)=e^x$

Neutrale Elemente: $\mathbb{Z} = 0 \rightarrow e1$

$\mathbb{R} = 1 \rightarrow e2$

B1: $f(a+b)=f(a)+f(b) \rightarrow e^{a+b} = e^a \cdot e^b$

B2: $f(e1)=e2 \rightarrow e^0 = 1$

1.14.Ordnen von Gruppenelementen (Engl.: Order of Elements)

Das Ordnen von Gruppenelementen wird wie folgt definiert: Die kleinste nat. Zahl e wird die Ordnung von Gruppenelementen genannt, wenn $g^e \equiv 1 \bmod m$. Anders herum gesagt, ist die Gleichung $g^e \equiv 1 \bmod m$ nur dann gültig, falls die natürliche Zahl e dividierbar durch die Ordnung von g ist.

Beispiel: Ordnung von Gruppenelementen $\rightarrow$ m(nat. Zahl)=13, g(Integer)=2

$g^1 \equiv 2 \bmod 13 \Rightarrow g^2 \equiv 4 \bmod 13 \Rightarrow g^3 \equiv 8 \bmod 13 \Rightarrow g^4 \equiv 16 \equiv 3 \bmod 13$

$g^5 \equiv 32 \equiv 6 \bmod 13 \Rightarrow g^6 \equiv g^5 \cdot g \equiv 6 \cdot 2 \equiv 12 \bmod 13$

$g^7 \equiv g^6 \cdot g \equiv 12 \cdot 2 \equiv 24 \equiv 11 \bmod 13 \Rightarrow g^8 \equiv g^7 \cdot g \equiv 11 \cdot 2 \equiv 22 \equiv 9 \bmod 13$

$g^9 \equiv g^8 \cdot g \equiv 9 \cdot 2 \equiv 18 \equiv 5 \bmod 13 \Rightarrow g^{10} \equiv g^9 \cdot g \equiv 5 \cdot 2 \equiv 10 \bmod 13$

$g^{11} \equiv g^{10} \cdot g \equiv 10 \cdot 2 \equiv 20 \equiv 7 \bmod 13 \Rightarrow g^{12} \equiv g^{11} \cdot g \equiv 7 \cdot 2 \equiv 14 \equiv 1 \bmod 13$

Order of Elements ist e=12 (mit Primitivwurzel g=2 mod 13)

1.15.Primitivwurzel (Engl.: Generator)

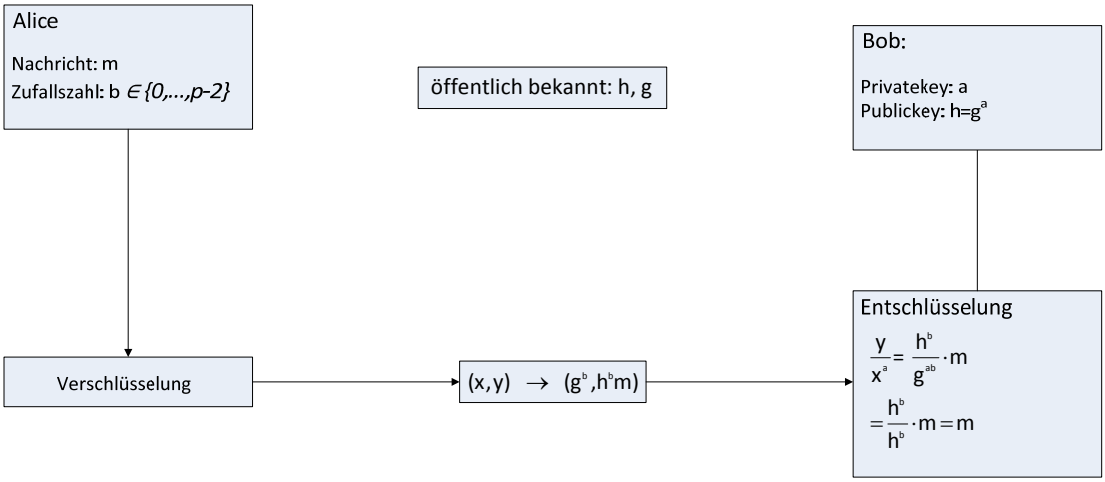
Eine ganze Zahl g, für die die Restklasse $g + p\mathbb{Z}$ die prime Restklassengruppe $(\mathbb{Z} / p\mathbb{Z})^*$ erzeugt, heisst Primitivwurzel mod p. Es sei p eine Primzahl. Somit ist ein Element $g \in \{1,2,\ldots, p-1\}$ eine Primitivwurzel von mod p, falls gilt: Menge$\{g^i \bmod p: 1 \leq i \leq p-1\}$.

Anders gesagt ist das Element g dann ein Generator, falls seine Ordnung mod p dem Wert p-1 entspricht: $g^e \bmod p \equiv 1 \bmod p$.

2. ElGamal Verschlüsselung

2.1. ElGamal Verschlüsselung

Die ElGamal Verschlüsselung hängt eng mit dem Diffie-Hellman-Schlüsselaustausch zusammen. Wie bei DH (im Jahre 1994 durch Ueli Maurer bewiesen), beruht auch bei ElGamal die Sicherheit der Verschlüsselung auf der Schwierigkeit, diskrete Logarithmen zu berechnen. Das Verfahren wurde im Jahre 1975 von Taher ElGamal entwickelt.



Schlüsselerzeugung:

1) User Alice wählt eine Primzahl p und eine Primitivwurzel g mod p.
2) Danach wählt Alice einen zufälligen und gleichverteilten Exponenten $a \in \{0, \dots, p-2\}$
3) Alice berechnet daraus $A = g^a \bmod p$
4) Der öffentliche Schlüssel von Alice ist somit (p,g,A)
5) Der geheime Schlüssel von Alice ist der Exponent a.

Verschlüsselung:

1) Der Klartextraum bei ElGamal ist dabei die Menge M={0, … , p-2}
2) Damit Bob den geheimen Text m verschlüsseln kann, besorgt er sich den öffentlichen Schlüssel von Alice (p,g,A)
3) Bob wählt danach eine Zufallszahl $b \in (0, \dots, p-2)$
4) Danach berechnet er $B = g^b \bmod p$
5) Am Schluss multipliziert Bob die Nachricht m mit $A^b \bmod p = g^{ab} \bmod p$ und erhält $c = A^b m \bmod p$
6) Bob sendet den Schlüsseltext (B,c) zu Alice

Entschlüsselung:

1) Um die Nachricht m zu erhalten, bestimmt Alice den Exponenten x=p-1-a

$B^x c \equiv g^{b(p-1-a)} A^b m \equiv (g^{p-1})^b (g^a)^{-b} A^b m \equiv A^{-b} A^b m \equiv m \bmod p$

(letzter Schritt mittels kleiner Satz von Fermat)

Beweis, das $(g^{p-1})^b (g^a)^{-b} A^b m \equiv A^{-b} A^b m$ gilt:

1) $A^b \bmod p \equiv g^{ab} \bmod p$

2) Little Fermat: $a^{p-1} \equiv 1 \bmod p$

Aus 1) und 2) folgt:

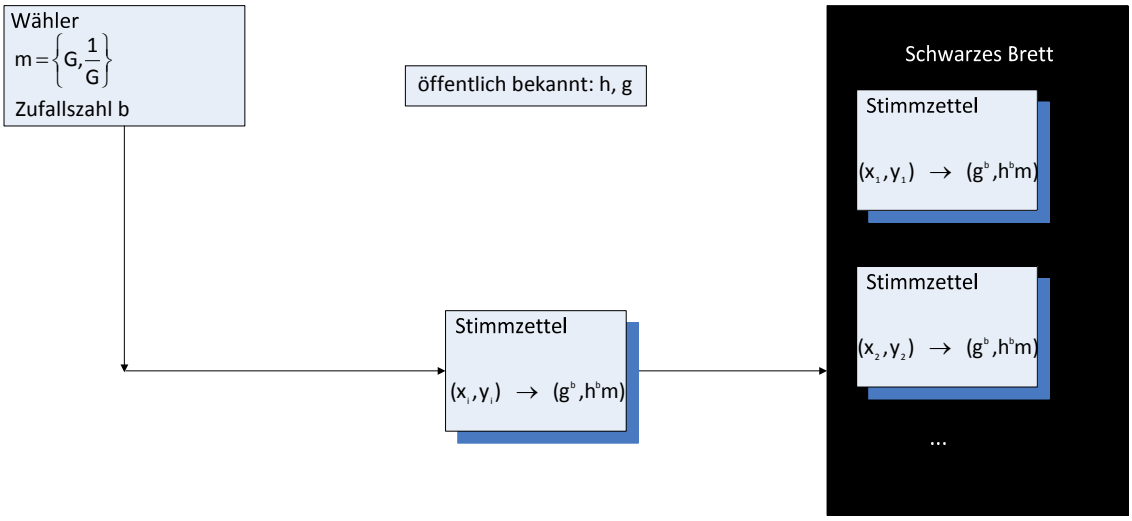
$$\underbrace{(g^{p-1})^b}_{1} (g^a)^{-b} A^b m \rightarrow (g^a)^{-b} A^b m \rightarrow g^{-ab} A^b m \rightarrow A^{-b} A^b m$$

2.2. Das diskrete Logarithmus Problem

In der Kryptographie ist der diskrete Logarithmus die Umkehrfunktion der diskreten Exponentiation. Dabei gilt die Berechnung des diskreten Logarithmus, ähnlich wie die Zerlegung von Zahlen in Primfaktoren, als schwer zu berechnen und bildet deshalb die Grundlage für die Sicherheit von gewissen kryptographischen Verfahren. Im Diffie-Hellman-Schlüsselaustausch bildet die diskrete Exponentiation die Grundlage für die Verschlüsselung $\rightarrow f : x \rightarrow \alpha^x \bmod p.$ Es wird allgemein angenommen, dass die diskrete Exponentiation eine Einweg-Funktion ist. Der Grund dazu liegt in der Schwierigkeit die Umkehrfunktion (den diskreten Logarithmus), also die Berechnung von x aus y, zu bestimmen $\rightarrow x = \log_\alpha (y).$ Bis jetzt gibt es noch keinen effizienten Algorithmus um dieses Problem zu lösen. Einige Ansätze zur Berechnung von solchen diskreten Logarithmen sind mit dem Baby-Step-Giant-Step oder dem Square-and-Multiply Verfahren gegeben.

2.3. E-Voting basierend auf dem ElGamal Kryptosystem

Ronald Cramer et al. [2] haben ein E-Voting Verfahren entwickelt, dass auf dem ElGamal Kryptosystem basiert. Das E-Voting Verfahren gebraucht dabei das ElGamal-Verschlüsselungsverfahren. Der Wähler verschlüsselt seine Stimme m (in diesem Beispiel entweder Ja / Nein) mit dem allgemein zugänglichen öffentlichen Schlüssel h und dem Generator g. Alle verschlüsselten Votes werden danach auf dem so genannten schwarzen Brett (bulletin board) aufgelistet. Dieses schwarze Brett ist öffentlich zugänglich und jedermann kann seine Stimme dort auf Integrität prüfen. Um das Wahlergebnis zu bestimmen, werden die gesammelten Stimmen des schwarzen Brettes mittels geeigneten Verfahren ausgezählt. Eines dieser Verfahren ist das so genannte homomorphic tallying. Eine andere Variante ist das Benützen von Mixnets (eine Art digitales Schütteln der Urne). Damit werden die Votes anonymisiert, sprich sie können nicht mehr dem einzelnen Wähler zugeordnet werden. Danach können sie mit geeigneten Verfahren ausgezählt werden. Natürlich sollte keine Einzelperson (auch kein Offizieller) alleine die Votes vom schwarzen Brett entschlüsseln können, da er sonst als Einzelperson die Wahl fälschen kann. Um dies zu verhindern, wird der private Schlüssel auf mehrere Wahlautoritäten aufgeteilt (=Secret Sharing).



2.4. Homomorphe Kryptographie

Der Grundgedanke bei homomorpher Kryptographie ist abgeleitet von der Grundeigenschaft des Homomorphismus: Die strukturerhaltende Abbildung. Es soll also möglich sein Nachrichten zu verschlüsseln, also abzubilden, aber die Eigenschaft der Nachricht (in unserem Falle der Wahlstimmeninhalt) strukturell beizubehalten. Es sollte also möglich sein Operationen auf die einzelnen verschlüsselten Nachrichten anzuwenden, ohne den Nachrichteninhalt entschlüsseln zu müssen. Diese Eigenschaft kann dann in der Auszählung von Stimmen gebraucht werden. Etwas formeller ausgedrückt ist eine Verschlüsselungsfunktion E homomorph, wenn für einen öffentlichen Schlüssel k gilt:

$$E_k(T_1 \oplus T_2) = E_k(T_1) \otimes E_k(T_2) \quad \rightarrow \text{ Dabei sind } \oplus, \otimes \text{ beliebige Gruppenoperationen}$$

Die Wahlautorität kann dann $c = \otimes E_k(M_i)$ und daraus $t = E_{k^{-1}}(c)$ berechnen. Im Falle von E-Voting ist die Gruppenoperation $\oplus$ vorzugsweise eine additive Funktion, damit die einzelnen Stimmen auch zu einem Gesamtergebnis ausgezählt werden können. Es wird zwischen multiplikativen (ElGamal, RSA) und additiven (Paillier) homomorphen Kryptosystemen unterschieden.

2.5. Homomorphe Kryptographie mit ElGamal

Bei ElGamal ist die homomorphe Verschlüsselung nicht wie bei Pailler additiv, sondern multiplikativ:

$$E_k(m_1) \cdot E_k(m_2) = (g^{b_1}, g^{ab_1} \cdot m_1) \cdot (g^{b_2}, g^{ab_2} \cdot m_2) = (g^{b_1+b_2}, g^{ab_1+ab_2} \cdot m_1 \cdot m_2) = E_k(m_1 \cdot m_2)$$

$\rightarrow$ Nicht additiv!

Da aber für eine Stimmenauszählung nur Addition in Frage kommt, muss die originale ElGamal Verschlüsselung angepasst werden. Dies wird erreicht, indem ein zusätzlicher öffentlicher Parameter eingesetzt wird: Generator G ($\neq$g). Das Resultat dieser Modifikation ist, dass die Nachricht, in diesem Fall die Stimme, als Exponent verwendet wird und die eigentliche Verschlüsselung mit ElGamal im Exponenten stattfindet. Dies führt dazu, dass am Schluss wie bei Pailler ein additiver Homomorphismus herrscht:

$$E_k(m_1)\cdot E_k(m_2)=(g^{b_1},g^{ab_1}\cdot G^{m_1})\cdot(g^{b_2},g^{ab_2}\cdot G^{m_2})=(g^{b_1+b_2},g^{ab_1+ab_2}\cdot G^{m_1\cdot m_2})=E_k(m_1+m_2)$$

Durch das Einführen dieses Generators G ergibt sich auch die grösste Schwäche und Gefahr der homomorphen Kryptographie mit dem ElGamal Verfahren: Das diskrete Logarithmus Problem.

2.6. Diskreter Logarithmus Problem mit homomorphen ElGamal Kryptoverfahren

Das ElGamal Kryptosystem hat eine natürliche multiplikative homomorphe Eigenschaft:

Für alle $i \in \{1,2,...,n\}$ werden n Ciphertexte zu folgender Gleichung kombiniert:

$$\prod_{i=1}^{n} c_i = ((\prod_{i=1}^{n} \alpha_i),(\prod_{i=1}^{n} \beta_i))$$

Die Entschlüsselung resultiert in einer Kombination der individuellen Nachrichten, wobei x den privaten Entschlüsselungs-Schlüssel darstellt:

$$\prod_{i=1}^{n} s_i = \frac{\prod_{i=1}^{n} \beta_i}{(\prod_{i=1}^{n} \alpha_i)^x}$$

Will man nun aus der multiplikativen Eigenschaft eine additive machen, muss das ElGamal Verfahren wie schon erwähnt modifiziert werden. Es wird ein genereller Generator g_1 für die Gruppe G verwendet. Ein Ciphertext s wird verschlüsselt, indem ein zufälliger Wert $r \in \mathbb{Z}_q$, g_1 gewählt wird, so dass gilt [3]:

$c = (\alpha,\beta) = (g^r, g_1^s y^r)$, wo $y=g^x$ und g auch ein öffentlich zugänglicher Generator der Gruppe G ist.

Die Entschlüsselung kann nur durch die Findung eines diskreten Logarithmus (DL) erfolgen und resultiert in einer Summation der einzelnen Nachrichten:

DL= Diskreter Logarithmus Suchfunktion

$$DL=(\frac{\prod_{i=1}^{n} \beta_i}{(\prod_{i=1}^{n} \alpha_i)^x})=DL(\prod_{i=1}^{n} g_1^{s_i})=DL(g_1^{\sum_{i=1}^{n} s_i})=\sum_{i=1}^{n} s_i$$

Diese Suche nach einem diskreten Logarithmus kann je nach Suchraum (Anzahl der Ciphertexts, was nichts anders als die Anzahl Stimmen sind) rechnerisch sehr anspruchsvoll sein. Diese Suche muss mit heutzutage bekannten Algorithmen wie Baby-Step-Giant-Step o.a. durchgeführt werden. Dieser Umstand ist ein grosser Nachteil des ElGamal-Verfahrens, da z.T andere Verfahren wie zum Beispiel das Pailler-Kryptosystem, von Natur aus additiv homomorph sind. Die Rechenintensität kann aber noch verringert werden, indem die Ciphertexte in kleineren Gruppen gruppiert werden.

2.7. Homomorphic Tallying

Mit homomorphic tallying ist es möglich verschlüsselte Werte in einen dritten, ebenfalls verschlüsselten Wert zu verwandeln. Dieser dritte Wert hat dabei eine Beziehung zu den 2 vorherigen Werten, z.B kann er die Summe der Beiden sein. Man kann dabei die verschlüsselten Werte am Anfang nicht einzeln entschlüsseln und anschauen (Integrität). Als vereinfachtes Beispiel im Bereich E-Voting kann man sich z.B eine Wahl mit Wahlmöglichkeiten Ja (=0) / Nein (=1) angeschaut werden:

1) Eine gültige Stimme $m \in \{0,1\}$
2) Diese werden verschlüsselt und homomorphisch addiert
3) Das Ergebnis (z.B 10) ist die Anzahl Nein Stimmen. Die Anzahl Ja- Stimmen wird aus der Gesamtzahl der Stimmen abzüglich der Nein-Stimmen bestimmt

Als anderes Beispiel kann man sich z.B auch eine Wahl mit den Wahlmöglichkeiten Ja (1) / Nein (-1) vorstellen. Als Ergebnis der Entschlüsselung erhält man danach entweder ein positives (mehr 1 denn -1 Stimmen), oder ein negatives (mehr -1 denn 1 Stimmen) Endresultat. Falls das Resultat positiv ist, hat Ja gewonnen, bei einem negativen Endergebnis entsprechend Nein.

$p = 11$, Generator g=6, G=8, $G^{-1} = 7$, Privatekey a=2, Publickey $h=g^a = 3$

3 Wähler:

B1= zufälliger Generator b=9 $\rightarrow$ Ja-Stimme $\rightarrow E_k(m_1) = (g^b, g^{ab} \cdot G^m) = (2,-1)$

B2= zufälliger Generator b=4 $\rightarrow$ Nein-Stimme $\rightarrow E_k(m_2) = (g^b, g^{ab} \cdot G^m) = (9,6)$

B3= zufälliger Generator b=3 $\rightarrow$ Nein-Stimme $\rightarrow E_k(m_3) = (g^b, g^{ab} \cdot G^m) = (7,2)$

$(X,Y) = (2 \cdot 9 \cdot 7,\ -1 \cdot 6 \cdot 2) \equiv (5,10)$

$Y \cdot X^{-s} \equiv 7 \equiv G^T \rightarrow$ T=-1 Das Ja/Nein Verhältnis

3. Homomorphic Tallying mit ElGamal - Probleme

3.1. Baby-Step-Giant-Step

Der Babystep / Giantstep Algorithmus kann den diskreten Logarithmus eines Elements einer Gruppe berechnen. Er ist der Brute-Force Methode weit überlegen, kann aber für sehr grosse Gruppen bisher noch keine Berechnung in annehmbarer Zeit liefern. Ein Beispiel:

g=11, a=3, 11 ist die Primitivwurzel mod 29, $n=\{\frac{\mathbb{Z}}{29\mathbb{Z}}\}^*=28$

Gesucht ist x, so dass: $x=\log_{11} 3$ in $\{\frac{\mathbb{Z}}{29\mathbb{Z}}\}^*$, also $11^x \equiv 3 \bmod 29$

1) Setze $m=\sqrt{n} = 6$

2) Ziel: Zerlege x in seine eindeutige Darstellung: x=im+j (man bestimmt also i und j)

3) Giant steps: Bestimme die Paare (j,g^j) für $0 \leq j<m$

j	0	1	2	3	4	5
11^j	1	11	5	26	25	14

4) Baby steps: Bestimme die Paare $(i,a(g^{-m})^i)$ für $0 \leq j<m$ und terminiere, falls $a(g^{-m})^i = g^j$

i	0	1	2
$3*(13)^i$	3	10	14

Somit ist mit $11^{-6} \equiv (11^6)^{-1} \equiv 13$

5) Die Lösung ist dann x=im+j, denn $a(g^{-m})^i = g^j \rightarrow a=g^{im+j} = g^x$

$x = 2 \cdot 6 + 5 = 17$

4. Schlussfolgerung

Das ElGamal Kryptosystem ist eine geeignete Methode um bei E-Voting Systemen zum Einsatz zu kommen. Der Nachteil der multiplikativen homomorphischen Eigenschaft wiegt aber relativ schwer, vor allem weil es andere kryptographische Systeme mit natürlichen additiven homomorphen Eigenschaften gibt. Ebenfalls kann die Rechenintensität bei grossem Wähleranteil wie z.B der USA schnell zu einem Problem werden.

5. References

[1] Wikipedia. (2009). Homomorphismus, [Online], Available: http://en.wikipedia.org/wiki/Homomorphismus [Dez 2009].

[2] Cramer, R. et al. A secure and optimal efficient multi-authority election scheme, [Online], Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.8654 [Dez 2009].

[3] Buchmann, J. (2008). Einführung in die Kryptographie, vierte Auflage. Berlin: Springer-Verlag.

[4] Ondrisek, B. Dissertation: Sicherheit elektronischer Wahlen , [Online], Available: http://www.seres-unit.com/uploads/arbeit_barbara_ondrisek.pdf [Dez 2009].

[5] Adida, B. Advances in cryptographic voting systems, [Online], Available: http://theory.lcs.mit.edu/~cis/theses/adida-phd.pdf [Dez 2009].

[6] Aditya, R. Secure electronic voting with flexible ballot structure, [Online], Available: http://eprints.qut.edu.au/16156/1/Riza_Aditya_Thesis.pdf [Jan 2010].