Universität Mannheim, Reihe Informatik, TR-2006-xxx

Kryptowochenende 2006 — Workshop über Kryptographie
Universität Mannheim

Frederik Armknecht
NEC Europe Ltd.

Dirk Stegemann
Universität Mannheim

01.–02. Juli 2006

Inhaltsverzeichnis

Public-Key-Kryptographie mit Halbgruppen-Aktionen und Halbringen

Jens Zumbrägel

Institut für Mathematik, Universität Zürich
Winterthurerstr. 190, CH - 8057 Zürich
jzumbr@math.unizh.ch

Das klassische Diffie-Hellman-Protokoll zum Schlüsselaustausch kann im Kontext von Halbgruppen-Aktionen verallgemeinert werden [MMR05]: Öffentlich gegeben sei hierzu eine Halbgruppen-Aktion, d.h. eine (effizient berechenbare) Abbildung

$$G \times S \to S, \quad (g, s) \mapsto g.s$$

einer endlichen abelschen Halbgruppe $(G, \cdot)$ auf eine endliche Menge S, so dass die Identität $g.(h.s) = (g \cdot h).s$ für alle $g, h \in G$ und $s \in S$ erfüllt ist. Desweiteren soll für $s \in S$ die Bahnen-Abbildung $G \to G.s \subseteq S$, $g \mapsto g.s$ eine Einweg-Funktion sein, so dass also die Schwierigkeit des folgenden Problems besteht:

$$\text{gegeben } s \in S \text{ und } t \in G.s, \text{ finde ein } g \in G \text{ mit } t = g.s.$$

Wir haben dann die folgende Abwandlung des Diffie-Hellman-Protokolls:

Alice	öffentlich	Bob
	wähle $s \in S$	
wähle $a \in G$	$a.s \in S$	
	$b.s \in S$	wähle $b \in G$
berechne $a.(b.s)$		berechne $b.(a.s)$

Alice und Bob besitzen somit den gemeinsamen Schlüssel $(a \cdot b).s$.

Besitzt S zusätzlich die Struktur einer Gruppe, so ergibt sich außerdem eine Verallgemeinerung des ElGamal-Protokolls zur Chiffrierung.

Im klassischen Fall haben wir hierbei als Aktion die Potenz-Abbildung $\mathbb{Z} \times H \to H$, $(n, a) \mapsto a^n$ der ganzen Zahlen $(\mathbb{Z}, \cdot)$ auf eine Gruppe $(H, \cdot)$, wobei H z.B. die multiplikative Gruppe $\mathbb{F}^*$ eines endlichen Körpers $\mathbb{F}$ oder die Gruppe $E(\mathbb{F})$ der $\mathbb{F}$-rationalen Punkte einer elliptischen Kurve E ist.

Aus verschiedenen Gründen ist es interessant, verallgemeinerte Halbgruppen-Aktionen zu konstruieren und zu analysieren. Als wichtige Bausteine für die Konstruktion von Halbgruppen-Aktionen erweisen sich sogenannte Halbringe. Unter einem *Halbring* versteht man einen „verallgemeinerten Ring“ in denen keine negativen Elemente zu existieren brauchen, d.h.

$$(R, +, \cdot) \text{ ist Halbring } \Leftrightarrow \begin{cases} (R, +) \text{ ist abelsche Halbgruppe,} \\ (R, \cdot) \text{ ist Halbgruppe,} \\ \text{Distributivgesetze gelten.} \end{cases}$$

Elemente $0, 1 \in R$ mit $0 + x = x$, $0x = 0 = x0$ bzw. $1x = x = x1$ für alle $x \in R$ werden Null bzw. Eins genannt. Beispiele für Halbringe mit Null und Eins sind die natürlichen Zahlen $\{0, 1, 2, \dots\}$ mit gewöhnlicher Addition und Multiplikation, sowie der *boolesche Halbring* $(S, \texttt{or}, \texttt{and})$, wobei $S := \{0, 1\}$ und `or` bzw. `and` die entsprechenden logischen Operationen sind.

Um eine Pohlig-Hellman-artige Reduktion auf kleinere Instanzen zu vermeiden, ist es außerdem wichtig, dass der Halbring R im folgenden Sinne *einfach* bzw. *kongruenz-frei* ist: Jede Äquivalenzrelation $\sim$ auf R, die

$$a \sim b,\ c \sim d \ \Rightarrow \begin{cases} a + c \sim b + d \\ a \cdot c \sim b \cdot d \end{cases}$$

für alle $a, b, c, d \in R$ erfüllt, ist trivial, d.h. $\sim = R \times R$ oder $\sim = \mathrm{id}_R$. Sicherlich sind alle Halbringe der Größe 2, wie der boolsche Halbring, einfach.

Die Klassifikation endlicher, kongruenz-freier Halbringe ist ein interessantes Problem, welches weiterhin offen ist (siehe [Mo04] für eine Teilklassifikation). Jedoch lässt sich eine unendliche Familie solcher Halbringe angeben, denn für jeden einfachen Halbring R mit Null und Eins ist der Matrix-Halbring $\mathrm{Mat}_{n\times n}(R)$ ebenfalls einfach.

Eine computer-unterstützte Suche meinerseits nach einfachen Halbringen brachte folgendes Ergebnis: Unter allen Halbringen R mit Null, die keine Ringe sind, und eine Größe $\#R \in \{3, \ldots, 12\}$ besitzen, gibt es bis auf Isomorphie nur einen einzigen einfachen Halbring. Dieser hat 6 Elemente $R = \{0, 1, \mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}$ und die folgenden Operationstafeln:

+	0	a	b	c	1	d
0	0	a	b	c	1	d
a	a	a	b	c	1	d
b	b	b	b	1	1	d
c	c	c	1	c	1	d
1	1	1	1	1	1	d
d	d	d	d	d	d	d

·	0	a	b	c	1	d
0	0	0	0	0	0	0
a	0	0	0	a	a	b
b	0	a	b	a	b	b
c	0	0	0	c	c	d
1	0	a	b	c	1	d
d	0	c	d	c	d	d

Es gibt also Indizien dafür, dass einfache Halbringe mit Null überaus selten sind.

Hat man nun einen einfachen Halbring R und zusätzlich einen R-Halbmodul M gegeben, so ergibt sich eine Halbgruppen-Aktion aus der Matrix-Multiplikation

$$\mathrm{Mat}_{n\times n}(R) \times M^n \to M^n;$$

hierbei wird dann eine abelsche Unter-Halbgruppe $G \subseteq \mathrm{Mat}_{n\times n}(R)$ gewählt. In diesem Kontext ergibt sich das klasssiche Diffie-Hellman-System als Spezialfall ($n = 1$). Ich werde Angriffsmöglichkeiten auf solche Systeme diskutieren, sowie eine weitere Halbgruppen-Aktion vorstellen, die Halbringe verwendet.

Literatur

[MMR05] Gérard Maze, Chris Monico, Joachim Rosenthal. Public Key Cryptography based on Semigroup Actions *Preprint*, arXiv:cs.CR/0501017v2 28. Januar 2005

[Mo04] C. Monico. On finite congruence-simple semirings *Journal of Algebra* **271**(2), 846–854, 2004

Äquivalente Schlüssel in $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Public Key Systemen — Aktueller Stand

Christopher Wolf

École Normale Supérieure, Département d'Informatique
45 rue d'Ulm, F-75230 Paris Cedex 05, France
Christopher.Wolf@ens.fr or chris@Christopher-Wolf.de

1 Initial Considerations

In the last 20 years, several schemes based on the problem of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations (or $\mathcal{MQ}$ for short) have been proposed. The most important ones certainly are MIA / C* and Hidden Field Equations (HFE) plus their variations MIA- / C^{*--}, HFE-, HFEv, and HFEv-. Both classes have been used to construct signature schemes for the European cryptography project NESSIE, namely the MIA- variation in Sflash, the HFEv- variation in Quartz and the HFE- variation in the tweaked version Quartz-7m. Unbalanced Oil and Vinegar schemes and Stepwise Triangular Schemes are also important in practice. While the first is secure with the correct choice of parameters, the second forms the basis of nested constructions like the enhanced TTM, Tractable Rational Maps, or Rainbow. An overview of all these systems can be found in the taxonomy article [WPc].

In this talk, we give an overview on the question of equivalent keys of $\mathcal{MQ}$-schemes. At first glance, this question seems to be purely theoretical. But for practical applications, we need memory and time efficient instances of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. One important point in this context is the overall *size* of the private key: in restricted environments such as smart cards, we want it as small as possible. Hence, if we can show that a given private key is only a representative of a much larger class of equivalent private keys, it makes sense to compute (and store) only a normal form of this key. Similar, we should construct new $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic schemes such that they do not have a large number of equivalent private keys but only a small number, preferably only one per equivalence class. This way, we make optimal use of the randomness in the private key space and neither waste computation time nor storage space without any security benefit.

All systems based on $\mathcal{MQ}$-equations use a public key of the form

$$p_i(x_1, \ldots, x_n) := \sum_{1 \le j \le k \le n} \gamma_{i,j,k} x_j x_k + \sum_{j=1}^{n} \beta_{i,j} x_j + \alpha_i \,,$$

with $n \in \mathbb{Z}^+$ variables and $m \in \mathbb{Z}^+$ equations. Moreover, we have $1 \le i \le m; 1 \le j \le k \le n$ and $\alpha_i, \beta_{i,j}, \gamma_{i,j,k} \in \mathbb{F}$ (constant, linear, and quadratic terms). We write the set of all such systems of polynomials as $\mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$. Moreover, the private key consists of the triple $(S, \mathcal{P}', T)$ where $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n), T \in \mathrm{Aff}^{-1}(\mathbb{F}^m)$ are bijective affine transformations. Moreover, we have $\mathcal{P}' \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ is a polynomial-vector $\mathcal{P}' := (p'_1, \ldots, p'_m)$ with m components; each component is a polynomial in n variables $x'_1, \ldots, x'_n$. Throughout this paper, we will denote components of this private vector $\mathcal{P}'$ by a prime $'$. In contrast to the public polynomial vector $\mathcal{P} \in \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m)$, the private polynomial vector $\mathcal{P}'$ does allow an efficient computation of $x'_1, \ldots, x'_n$ for given $y'_1, \ldots, y'_m$. Still, the goal of $\mathcal{MQ}$-schemes is that this inversion should be hard if the public key $\mathcal{P}$ alone is given. The main difference between $\mathcal{MQ}$-schemes lies in their special construction of the central

equations $\mathcal{P}'$ and consequently the trapdoor they embed into a specific class of $\mathcal{MQ}$-problems. An introduction to $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems is given in [WPc].

This talk is based on the two conference papers [WPa,WPb], which deal with the classes MIA, HFE, and UOV. An extended version which also includes STS and shows that the reduction for MIA/MIA for $q \neq 2$ is tight is [WPd].

2 Mathematical Considerations

Before discussing concrete schemes, we start with some general observations and definitions. Obviously, the most important term in this article is "equivalent private keys". We give a graphical

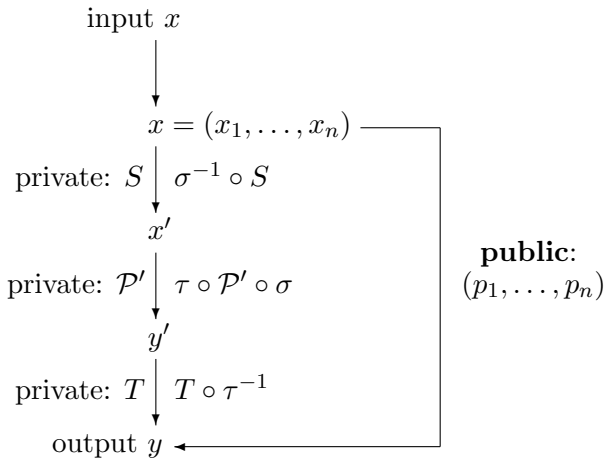

Figure 1: Equivalent private keys using affine transformations σ, τ

representation of this idea in Figure 1. We can also express this idea in the following definition:

DEFINITION 2.1 *We call two private keys*

$$(S, \mathcal{P}', T), (\tilde{S}, \tilde{\mathcal{P}}', \tilde{T}) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathit{Aff}^{-1}(\mathbb{F}^m)$$

"equivalent" if they lead to the same public key, i.e., if we have

$$T \circ \mathcal{P}' \circ S = \mathcal{P} = \tilde{T} \circ \tilde{\mathcal{P}}' \circ \tilde{S}\,.$$

In the above definition, $\mathrm{Aff}^{-1}(\cdot)$ denotes the class of bijective affine transformations. In order to find equivalent keys, we consider the following transformations:

DEFINITION 2.2 *Let $(S, \mathcal{P}', T) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathcal{MQ}(\mathbb{F}^n, \mathbb{F}^m) \times \mathit{Aff}^{-1}(\mathbb{F}^m)$, and consider the four transformations $\sigma, \sigma^{-1} \in \mathit{Aff}^{-1}(\mathbb{F}^n)$ and $\tau, \tau^{-1} \in \mathit{Aff}^{-1}(\mathbb{F}^m)$. Moreover, let*

$$\mathcal{P} = T \circ \tau^{-1} \circ \tau \circ \mathcal{P}' \circ \sigma \circ \sigma^{-1} \circ S\,. \tag{1}$$

We call the pair $(\sigma, \tau) \in \mathit{Aff}^{-1}(\mathbb{F}^n) \times \mathit{Aff}^{-1}(\mathbb{F}^m)$ "sustaining transformations" for an $\mathcal{MQ}$-system if the "shape" of $\mathcal{P}'$ is invariant under the transformations σ and τ. For short, we write $(\sigma, \tau) \bullet (S, \mathcal{P}', T)$ for (2.2) and (σ, τ) sustaining transformations. This idea has already been outlined in Figure 1.

3 Sustaining Transformations

We have several sustainers which can be used with different multivariate quadratic public key systems.

Additive Sustainer: Add a constant $A \in \mathbb{E}$ or $a \in \mathbb{F}^n, b \in \mathbb{F}^m$.

Big Sustainer: Multiply with a non-zero constant $B \in \mathbb{E}^*$.

Small Sustainer: Multiply with a diagonal matrix with non-zero coefficients $b_1, \ldots, b_n, b'_1, \ldots, b'_m \in \mathbb{F}^*$, respectively.

Permutation Sustainer: Permute the input variables / the equations.

Gauss Sustainer: Perform Gauss operations.

Frobenius Sustainer: Perform the operation $X \to X^{q^i}$ for $1 \leq i \leq n$ and $i \in \mathbb{N}$.

Reduction Sustainer: Observe that the last r rows have no effect with $r \in \mathbb{N}$ being the number of equations missing.

These sustainers can now be combined with different multivariate quadratic public key systems. We summarise their effects in the next section.

4 Results

The sustainers outlined above can be applied to several basic classes, such as Hidden Field Equations (HFE), Matsumoto-Imai Scheme A (MIA), Unbalanced Oil and Vinegar schemes (UOV), and

Table 1: Summary of the reduction results of this article

Scheme	Reduction
UOV	$q^{n+mn} \prod_{i=0}^{n-m-1}(q^{n-m} - q^i) \prod_{i=0}^{m-1}(q^m - q^i)$
STS	$q^{m+n} \prod_{i=1}^{L} \left(q^{n_i(n-\sum_{j=1}^{i} n_j)} \prod_{j=0}^{n_i-1}(q^{n_i} - q^j) \right)$ $\prod_{i=1}^{L} \left(q^{m_i(n-\sum_{j=1}^{i} m_j)} \prod_{j=0}^{m_i-1}(q^{m_i} - q^j) \right)$
MIA	$n(q^n - 1)$
MIA-	$n(q^n - 1)q^r \prod_{i=n-r-1}^{n-1}(q^n - q^i)$
HFE	$nq^{2n}(q^n - 1)^2$
HFE-	$nq^{2n}(q^n - 1)(q^{n-r} - 1) \prod_{i=n-r-1}^{n-1}(q^n - q^i)$
HFEv	$n'q^{n+n'+vm}(q^{n'} - 1)^2 \prod_{i=0}^{v-1}(q^v - q^i)$
HFEv-	$n'q^{r+2n'vn'}(q^{n'} - 1)^2 \prod_{i=0}^{v-1}(q^v - q^i) \prod_{i=n'-r-1}^{n'-1}(q^{n'} - q^i)$

Stepwise-Triangular Systems (STS). We have summarised our results in tables 1 and 2. The first gives an overview on the formulae achieved while the latter features some numerical examples. The

symbols used in Table 1 are defined as follows: $n \in \mathbb{Z}^+$ denotes the number of variables, $m \in \mathbb{Z}^+$ is the number of equations, $q := |\mathbb{F}|$ is the number of elements in the ground field $\mathbb{F}$, L the number of layers for STS, and n_l, m_l for $1 \leq l \leq L$ the number of new variables and equations, respectively.

Table 2: Numerical examples for the reduction results of this article

Scheme	Parameters	Choices for S, T (in $\log_2$)	Reduction (in $\log_2$)
UOV	$q = 2, m = 64, n = 192$	37,054	32,956
	$q = 2, m = 64, n = 256$	65,790	57,596
STS	$q = 2, r = 4, L = 25, n = 100$	20,096	11,315
	$q = 2, r = 5, L = 20, n = 100$	20,096	11,630
HFE	$q = 2, n = 80$	12,056	326
HFE-	$q = 2, r = 7, n = 107$	23,108	2129
HFEv	$q = 2, v = 7, n = 107$	21,652	1160
HFEv-	$q = 2, r = 3, v = 4, n = 107$	22,261	1258
MIA	$q = 128, n = 67$	63,784	469
MIA-	$q = 128, r = 11, n = 67$	63,784	6180

We see applications of our results in different contexts. First, they can be used for memory efficient implementations of the above schemes: instead of saving the whole private key, we can only save a normal form. Second, they apply to cryptanalysis as they allow to concentrate on special forms of the private key. Third, constructors of new schemes should keep these sustaining transformations in mind: there is no point in having a large private key space — if it can be reduced immediately by an attacker who can just apply some sustainers. Moreover, the results obtained in this talk shine new light on cryptanalytic results, in particular key recovery attacks: as each private key is only a representative of a larger class of equivalent private keys, each key recovery attack can only recover it up to these equivalences as the public key $\mathcal{P}$ cannot contain information about individual private keys but the equivalence class used to construct $\mathcal{P}$.

References

[WPa] Christopher Wolf and Bart Preneel. Superfluous keys in $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic asymmetric systems. In *Public Key Cryptography — PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*. pages 275–287. Serge Vaudenay, editor, Springer, 2005. Extended version `http://eprint.iacr.org/2004/361/`.

[WPb] Christopher Wolf and Bart Preneel. Equivalent keys in HFE, C*, and variations. In *Proceedings of Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 33–49. Serge Vaudenay, editor, Springer, 2005. Extended version `http://eprint.iacr.org/2004/360/`, 15 pages.

[WPc] Christopher Wolf and Bart Preneel. Taxonomy of public key schemes based on the problem of multivariate quadratic equations. Cryptology ePrint Archive, Report 2005/077, 12$^{\text{th}}$ of May 2005. `http://eprint.iacr.org/2005/077/`, 64 pages.

[WPd] Christopher Wolf and Bart Preneel. Equivalent Keys in Multivariate Quadratic Public Key Systems. Cryptology ePrint Archive, Report 2005/464, 22$^{\text{nd}}$ of December 2005. `http://eprint.iacr.org/2005/464/`, 19 pages.

Effiziente Bestimmung der Algebraischen Immunität

Simon Künzli

Fachhochschule Nordwestschweiz
CH-5210 Windisch (Schweiz)
simon.kuenzli@fhnw.ch

1 Einleitung

Stromchiffren sind grundlegende kryptografische Algorithmen, um die Vertraulichkeit der elektronisch übermittelten Daten zu gewährleisten. Verglichen mit anderen Verfahren sind Stromchiffren sehr schnell, und eine Implementierung ist oft mit sehr wenig Hardware möglich. Daher sind Stromchiffren besonders geeignet in einer Umgebung mit wenig Ressourcen, etwa in kabellosen mobilen Netzwerken.

Wir betrachten eine Klasse von Stromchiffren, die auf einem linearen Schieberegister (LFSR) und einer Boolschen Filterfunktion basieren. Solche Verfahren haben einen geheimen N-bit Zustand x^t zur Zeit t. Der Zustand wird mit einem geheimen Schlüssel K initialisiert und mit einem LFSR L aktualisiert entsprechend $x^{t+1} = L(x^t)$. Eine nichtlineare boolesche Filterfunktion f mit n Variablen und algebraischem Grad k wird auf den Zustand angewendet, um ein Schlüsselstrom-bit $z^t = f(x^t)$ zu erzeugen. Ein Klartext-bit p^t wird dann zum Chiffretext-bit $c^t = p^t \oplus z^t$.

2 Angriffe

Bei einem Angriff auf LFSR-basierte Stromchiffren wird angenommen, dass L und f bekannt sind, dass der Angreifer viele Schlüsselstrom-bits z^t kennt (Angriff mit bekanntem Klartext), und dass sein Ziel ist die Rekonstruktion von x^t (und damit K) ist.

Mit dem bekannten Algorithmus von Berlekamp und Massey kann der Schlüsselstrom synthetisiert werden, und dies mit einer Komplexität von etwa $\binom{N}{k}^2$. Inzwischen sind aber viele spezifische Verfahren beweisbar resistent gegen diese *Berlekamp-Massey-Synthese*.

In einem anderen Ansatz kann der Angreifer ein multivariates, nichtlineares und überbestimmtes Gleichungssystem $z^t = f(x^t)$ für verschiedene Zeiten t aufstellen. Das System kann etwa durch *Linearisierung* gelöst werden, wobei jeder nichtlineare Term durch eine neue Variable ersetzt wird. Schließlich wird das lineare Gleichungssystem mit bekannten Methoden gelöst (etwa durch Gauss'sche Elimination). Abhängig vom *algebraischen Grad* der Gleichungen werden durch Linearisierung viele zusätzliche Variablen eingeführt. Dabei ist der Grad der Gleichungen durch den Grad von f bestimmt, und der Aufwand für das Lösungsverfahren beträgt etwa $\binom{N}{k}^3$. Das ist natürlich nicht effizient, aber ein geschickter Angreifer kann Gleichungen mit reduziertem Grad aufstellen. Solche leistungsfähigen Angriffe sind seit wenigen Jahren bekannt als *algebraische Angriffe*.

3 Algebraische Angriffe

Wie ist es möglich, den Grad der Gleichungen zu reduzieren? Eine Idee ist es, eine geeignete Funktion g von tiefem Grad d zu suchen, so dass $f \cdot g = 0$ in GF(2). Eine solche Funktion g nennt man *Annihilator* von f. Die Gleichung $z^t = f(x^t)$ kann mit dem Annihilator $g(x^t)$ multipliziert

werden, und man erhält für $z^t = 1$ die Gleichung $g(x^t) = 0$ von Grad d. Ähnliche Gleichungen können mit einem Annihilator von $f + 1$ aufgestellt werden. Daher definiert man die *algebraische Immunität* einer Funktion f als minimalen Wert d, so dass $f \cdot g = 0$ oder $(f + 1) \cdot g = 0$ für eine Funktion g vom Grad d. Es ist bekannt, dass die algebraische Immunität einer Funktion f mit n Variablen höchstens $\lceil n/2 \rceil$ sein kann [CM03]. Mit bisherigen Algorithmen kann die algebraische Immunität einer beliebigen Funktion in $\mathcal{O}(D^3)$ bestimmt werden, wobei $D \approx \binom{n}{d}$. Mit einem neuen Algorithmus, der auf multivariater Polynominterpolation beruht, können wir die algebraische Immunität in nur $\mathcal{O}(D^2)$ bestimmen [ACGKMR06].

4 Schnelle Algebraische Angriffe

Als Reaktion auf diese Angriffe sind Klassen von Filterfunktionen konstruiert worden, die große (oder sogar maximale) algebraische Immunität aufweisen (sowie weitere kryptografisch wünschenswerten Eigenschaften). Allerdings haben sich auch die Angriffe weiter entwickelt und sind bekannt als *schnelle algebraische Angriffe* [C03]. Dabei werden Funktionen g und h von tiefem Grad gesucht, so dass gilt $f \cdot g = h$. Mit einem Aufwand, der von $d := \deg h$ abhängig ist, kann man Terme von Grad größer als $e := \deg g$ eliminieren, und Gleichungen von Grad e aufstellen. Dabei ist d mindestens so groß wie die algebraische Immunität von f, und e kann viel kleiner sein.

Es war eine offene Frage, ob Funktionen f mit guter (konventioneller) algebraischer Immunität auch gegenüber diesen neuen Angriffen resistent sind. Bekannte Algorithmen können so angepasst werden, dass die Immunität einer Funktion gegen schnelle algebraische Angriffe in $\mathcal{O}(D^3)$ bestimmt werden kann. Wir haben einen neuen, effizienten (und theoretisch fundierten) Algorithmus konstruiert, der diese Aufgabe für beliebige Funktionen in $\mathcal{O}(DE^2)$ durchführt, wobei $E \approx \binom{n}{e}$. Der Algorithmus basiert darauf, dass Gleichungen für g und h separiert werden können. Dann haben wir symmetrische Funktionen untersucht, da diese von speziellem Interesse für Hardware-Implementierungen sind. Die symmetrische Struktur konnte in einem noch effizienteren Algorithmus ausgeschöpft werden.

Nebst diesen allgemeinen Algorithmen zur effizienten Bestimmung der algebraischen Immunität einer Funktion haben wir auch spezifische Funktionen theoretisch und experimentell untersucht. Die Resultate lassen uns schließen, dass viele Klassen von Funktionen sehr verwundbar sind gegenüber schnellen algebraischen Angriffen, und dies trotz ihrer optimalen (konventionellen) algebraischen Immunität [ACGKMR06]. Diese Arbeit wurde zusammen mit den Universitäten Mannheim und Limoges und mit INRIA durchgeführt.

Literatur

[CM03] N. Courtois, and W. Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. In *Advances in Cryptology - EUROCRYPT 2003*, LNCS 2656, pages 345–359. Springer Verlag, 2003.

[C03] N. Courtois. Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In *Advances in Cryptology - CRYPTO 2003*, LNCS 2729, pages 176–194. Springer Verlag, 2003.

[ACGKMR06] F. Armknecht, C. Carlet, P. Gaborit, S. Künzli, W. Meier, and O. Ruatta. Efficient Computation of Algebraic Immunity for Algebraic and Fast Algebraic Attacks. In *Advances in Cryptology - EUROCRYPT 2006*, LNCS 4004, pages 147–164. Springer Verlag, 2006.

Konstruktion von booleschen Funktionen mit maximaler algebraischer Immunität

Hellen Altendorf
Universität Mannheim
Fakultät für Mathematik und Informatik
Lehrstuhl für Theoretische Informatik

Das Ziel dieser Arbeit ist es boolesche Funktionen $f : \{0,1\}^n \to \{0,1\}^m$ zu konstruieren, deren Graph $gr(f) = \{(x, f(x))|x \in \{0,1\}^n\} \subset \{0,1\}^{n+m}$ maximale algebraische Immunität besitzt.

Wir sagen die Funktion $p : \{0,1\}^n \to \{0,1\}$ annuliert eine Teilmenge $S \subset \{0,1\}^n$ (oder p ist Annihilator der Menge S), falls gilt $p(x) = 0\ \forall x \in S$. Die Algebraische Immunität $AI(S)$ einer Menge S ist definiert als das kleinstes $d \in \mathbb{N}$ für das ein nichttrivialer Annihilator p vom Grad d existiert.

Die konstruierten Funktionen maximaler Immunität sollen als Grundbausteine für symmetrische Blockchiffren dienen und so algebraische Angriffe auf das Kryptosystem erschweren.

Algebraische Angriffe auf Secret-Key Kryptosysteme bestehen daraus, nichttrivialen Annihilatoren von kleinem Grad für die Beziehung zwischen Input- und Outputbits zu finden und so ein Gleichungssystem von möglichst kleinem Grad aufzustellen und effizient zu lösen.

Spätestens seit dem hypothetischen Angriff auf den Advanced Encryption Standard (AES) von Courtois und Pieprzyk (2002) verlangt die Kryptographie nach besseren algebraischen Funktionen. Dieser Angriff ist zurückzuführen auf die geringe algebraischer Immunität der S-Boxen. Die Gleichungen sind hier quadratisch, obwohl die Formate der S-Boxen ($\{0,1\}^8 \to \{0,1\}^8$) eine Immunität von 3 gewähren könnten.

Konkrete Angriffe wurden erstmals 2003 von Courtois und Meier für eine spezielle Klasse von Schlüsselstromgeneratoren beschrieben. Dieser Angriff wurde von Armknecht und Krause (2003) auf eine allgemeinere Klasse erweitert, welche den E_0-Generator beinhaltet, der im Bluetooth Standard verwendet wird.

In dem Vortrag wird erläutert, wie man die maximale Immunität einer Funktion $f : \{0,1\}^n \to \{0,1\}^m$ bestimmt und wie man letztendlich mithilfe der Matroidtheorie Funktionen von maximaler Immunität findet.

Literatur

[AK] F. Armknecht and M. Krause. *Constructing Single- and Multi-Output Boolean Functions with Maximal Algebraic Immunity*, Proceedings of ICALP 2006.

Erste Erfahrungen zu meinem „Post-doc-Leben“ in der Industrie

Frederik Armknecht

Network Laboratories
NEC Europe Ltd.

Wie bei so vielen vor mir drängte sich im Laufe meiner Promotion die Frage auf, wie es danach weitergehen sollte. Klar war nur, dass ich immer viel Spass am Forschen hatte und diesem auch weiterhin nachgehen wollte. Dies impliziert normalerweise einen Berufsweg im akademischen Bereich, was auch lange Zeit meine Planung war. Dennoch bin ich schlussendlich in die Industrie gegangen, und zwar zum Forschunglabor von NEC in Heidelberg.

In meinem Vortrag möchte ich kurz darlegen, wie und warum ich meine Entscheidung gefällt habe, und wie meine bisherigen Eindrücke bezüglich Industrieforschung sind. Damit möchte ich allen Mut machen, auch alternative Berufswege in Betracht zu ziehen.

Analyse der Entwicklung von Malware

Oliver Schmid

Technische Universität Darmstadt

Als Fred Cohen im Jahr 1983 ein sich selbst reproduzierendes Programm namens *VD* vorstellte, wurde zum ersten Mal von einem Computervirus gesprochen. In den ersten Jahren entwickelte sich dieses Phänomen noch recht langsam und wurde erst 1987 zu einem echten Problem für die Computerwelt, als mit *Lehigh* der erste Virus gespeicherte Daten löschte. In den kommenden Jahren konnten die Programmierer mit nur wenigen innovativen Ideen aufwarten, und auch quantitativ hielt sich das Problem in Grenzen, so waren 1991 lediglich 300 verschiedene Exemplare bekannt. Makroviren, die sich wie *Concept* durch den Befall von Word-Dateien vermehrten, gewannen ab 1995 an Bedeutung, besonders als sie drei Jahre später erstmals wie *Tristate* beliebige Office-Dokumente infizieren konnten.

Das Internetzeitalter begann 1997, als sich die Schadprogramme erstmals über E-Mail, FTP und IRC replizierten und sich dadurch erheblich schneller verbreiten konnten, als es bisher durch die Infizierung von Disketten möglich war. Die ersten Epidemien verursachten 1999 der Makrovirus *Melissa* und im Folgejahr der VBS-Wurm *LoveLetter*. In den vergangenen Jahren haben sich Viren und Würmer nun immer schneller entwickelt und zeichnen sich dabei durch immer neue Eigenschaften aus. So haben sie beispielsweise ihre Tarnung verbessert, indem sie ihre Prozessnamen denen von Systemprogrammen und Antiviren-Software angleichen, außerdem sind sie in der Lage, Antiviren-Programme und Personal Firewalls zu deaktivieren sowie den Zugriff auf Webseiten für Sicherheitsupdates zu unterbinden.

Die Entwicklung des Phänomens Malware wird nun in zwei Analysen einer näheren Betrachtung unterzogen. Die Daten, welche den beiden Analysen zu Grunde liegen, wurden durch Auswertung der Virendatenbanken zweier Hersteller von Antiviren-Software gewonnen [SoA][SyA]. Dabei wurden nur die Schadprogramme in den Datensatz aufgenommen, die in einem der Monate des Beobachtungszeitraumes seit Januar 1998 zu den Top-Ten der gemeldeten Malware gehörten [SoT][SyT]. Die unterschiedlichen Eigenschaften der Viren und Würmer, wie beispielsweise Verbreitungsweg, Schadwirkung oder Methoden zur Tarnung, wurden dabei in 41 Variablen erfasst, ebenso wie die relative Auftrittshäufigkeit der erfassten Exemplare am Gesamtaufkommen an Malware.

Zunächst wurde eine Clusteranalyse durchgeführt, um festzustellen, wie schnell sich die Entwicklung der Malware vollzogen hat. Hierfür wurden für jedes Jahr die Eigenschaften der Schadprogramme nach ihrer relativen Auftrittshäufigkeit gewichtet und aufsummiert. Je kleiner nun der Abstand zweier Jahre in der Distanzmatrix ist, desto ähnlicher sind sich die in diesen beiden Jahren aufgetretenen Viren und Würmer.

Das Ergebnis zeigt, dass sich zwischen den Jahren 1998 und 1999, in denen Viren und Makroviren das Geschehen beherrschten, wenig veränderte. Es gab nur wenige neue Entwicklungen, wie beispielsweise den Makrovirus *Melissa*, der sich selbst über MS Outlook verschickte und damit eine neue Ära der Verbreitung einläutete. In den folgenden Jahren wurden die Schadprogramme immer vielseitiger, ihren größten Entwicklungssprung erlebten sie im Jahr 2001, als die Makroviren in die Bedeutungslosigkeit fielen und die E-Mail-Würmer gleich drei entscheidende Neuerungen vorweisen konnten: Erstmals fälschten sie ihren Absender und verbreiteten sich mittels einer eigenen SMTP-Engine an Adressen, welche sie auf der Festplatte ihres Opfers finden konnten.

Auch setzten sich in diesem Jahr zwei Entwicklungen des Vorjahres erst richtig durch, und zwar die Änderung von Registry-Einträgen durch die Malware und die Einrichtung von so genannten Backdoors, welche den Remote-Zugriff auf einen infizierten Rechner ermöglichen. Seit dem Jahr 2004 hingegen hat sich nicht mehr viel getan, zwar steigt die Zahl der Virendefinitionen weiterhin unbeirrt auf inzwischen über 100.000, es gab jedoch seither keine wirklichen Innovationen.

Im Anschluss erfolgte eine Faktorenanalyse, um zusammenhängende Variablen auf wenige Faktoren zu reduzieren, in diesem Fall konnten die 41 Variablen durch 7 Faktoren erklärt werden. Der Faktor mit dem größten Erklärungsgehalt fasst dabei die typischen Eigenschaften moderner E-Mail-Würmer zusammen, welche sich mit einer eigenen SMTP-Engine an Adressen verbreiten, die sie auf der Festplatte ihres Opfers finden und die Sprache ihres Mailtextes anhand des Domain-Kürzels des Empfängers auswählen. Diese Würmer zeichnen sich weiterhin dadurch aus, dass sie Einträge in der Registry ändern, eine Backdoor zum infizierten Rechner öffnen, und in ihrer Auftrittshäufigkeit über die Zeit hinweg kontinuierlich zugenommen haben.

Der zweite Faktor bildet die Gruppe der Trojaner ab, die gezielt Sicherheitslücken der Betriebssysteme von Microsoft ausnutzen, um auf den Rechner zu gelangen und Daten wie Kreditkartennummern oder Passwörter auszuspionieren. Schadprogramme, deren vornehmliches Ziel die Manipulation von Daten ist, werden durch den dritten Faktor erklärt, weitere Faktoren sind beispielsweise die Hardwareorientierung älterer Exemplare, die einen Interrupt im BIOS nutzen und sich im MBR der Festplatte einnisten, oder die unauffällige Verbreitung über Filesharing-Verzeichnisse und durch doppelte Dateiendungen.

Aus den Analysen ergibt sich eine deutliche Verbindung zwischen dem zeitlichen Verlauf und bestimmten Malware-Typen, so waren Viren und Makroviren bis 2000 stark vertreten, und wurden in den folgenden zwei Jahren von den Würmern abgelöst, die seit 2002 das Geschehen dominieren. Des Weiteren hat sich eine Entwicklung von der Vernichtung von Daten hin zu Datendiebstahl und dem Missbrauch der befallenen PCs als so genannte Bots gezeigt. Unmittelbares Phishing durch Malware ist jedoch entgegen verschiedener Pressemeldungen nicht verbreitet, vielmehr tritt eine Verbindung zwischen Malware und Phishing meist im Zusammenhang mit Spambots auf, die neben Werbeinhalten natürlich auch Phishing-Mails verschicken können.

Literatur

[SoA] http://www.sophos.com/virusinfo/analyses/

[SyA] http://securityresponse.symantec.com/avcenter/vinfodb.html

[SoT] http://www.sophos.com/security/top-10/

[SyT] http://www.symantec.com/region/de/PressCenter/virentrends.html

Jointly Generating Random Keys for the Fully Distributed Environment

Sebastian Faust* and Stefan Lucks†

* K.U.Leuven ESAT-COSIC
Kasteelpark Arenberg 10,
B-3001 Leuven-Heverlee, Belgium

† Universität Mannheim
Lehrstuhl fuer Theoretische Informatik
68131 Mannheim, Germany

Abstract

In this paper we introduce a new efficient method to jointly generate and share k random secret keys for discrete log based cryptosystems in a fully distributed environment between a group of parties $P = \{P_1, \ldots, P_n\}$. We call such a scheme a k joint random key generation *(k-JRKG)* protocol.

Compared with the well-known technique of distributed key generation, where the shared key is not known by any one party, the intention of a JRKG protocol is slightly different: every random key is known and shared by only one party. Here, our protocol guarantees the randomness of the keys under the DDH assumption. In particular, this applies to the keys of the corrupted parties. Hence, they do not have a chance to bias their keys to a non-uniform distribution.

Our protocol reduces the dominating factor for the communication complexity, the number of reliable broadcasts, by a factor of n compared with other approaches to this problem.

The security of our protocol can be proven for less than $(\frac{n}{2})$-corrupted parties under the DL-assumption in the random oracle model.

1 Introduction

In the current literature, protocols for jointly generating random keys are frequently used as building blocks in various protocols designed for a fully distributed environment like the internet [GJ04, GJKR96]. In particular the well-analyzed technique of distributed key generation [GJKR99, Ped91] has a wide area of application and allows often in the first place to formally prove the protocols' security.

Unfortunately, most distributed key generation protocols do suffer from high communication and computation complexity, hence limiting their useability in practice to small and static networks.

A first approach to overcome these drawbacks was presented by John Canny and Stephen Sorkin in 2004 [CS04]. Their idea relies on the fragmentation of the set of parties P in a network to reduce the size of the broadcast groups. However, this technique is only probabilistic, i.e., has a failure probability, needs a dealer to build up and manage the broadcast groups and finally requires that honest and corrupted parties are randomly distributed in P.

In our work we take a different stance. Rather than trying to develop an efficient generally applicable solution for doing distributed key generation, we focus on specific variants of DKG to reduce their communication complexity. Such a specific variant is presented in this paper by the k-JRKG protocol as an efficient solution for the generation of k random secrets, where each secret is known by only one party. This technique is applicable, for example, in the setup phase of the protocols described by Golle and Juels in [GJ04]. In particular, it decreases their communication costs compared to a trivial solution and altogether provides a more natural approach to fulfill the needed requirements.

At first glance the generation of $k \gg 2$ uniformly distributed secrets $x_1, \ldots, x_k$ and corresponding public values $y_i = g^{x_i}$, where each x_i is known by only one party, seems to be an application

for common verifiable secret sharing *(VSS)* schemes. There, a dealer chooses a private key x and shares it in a verifiable manner with the participants in P according to the mechanisms of a threshold scheme. Although this method is very efficient, it cannot be used for our purposes, because corrupted dealers can choose their private keys non-randomly, hence, contradicting the proof of security in [GJ04]. Therefore the authors propose to use the DKG protocol of Gennaro et al. *(GJKR-DKG)*, which indeed guarantees that all keys are uniformly distributed, but unfortunately decreases the efficiency.

In contrast to this, our protocol is more efficient in terms of communication complexity in that it reduces the number of broadcasts by a factor of n, but still guarantees the randomness of the keys. In particular, we are able to show that the randomness of all keys, including the keys of corrupted parties, is guaranteed under the DDH assumption. Furthermore, we prove the security of the protocol for less than $(\frac{n}{2})$-corrupted parties under the DL-assumption in the random oracle model.

2 The k-JRKG protocol

In general, the protocol is structured into two phases: An initial phase which is executed only once at the beginning of the protocol and a key generation (KG) phase which has to be repeated for every needed key. In particular, in the initial phase the parties $P = \{P_1, \ldots, P_n\}$ perform two instances of the GJKR-DKG scheme and use in the KG phase the ElGamal encryption and Feldman VSS.

In the following let p, q be two large, odd primes with $p = 2q + 1$. Let $g \in \mathbb{Z}_p^*$ be an element with order q and $<g>= G \subseteq \mathbb{Z}_p^*$ denote the subgroup of quadratic residues in $\mathbb{Z}_p^*$ generated by g for which the DDH assumption holds. Finally, let $F : \mathbb{Z}_q \mapsto G$ be an efficiently invertible bijection.

I. Initialization (executed only once):

1. The parties in P execute an instance of the GJKR-DKG scheme to generate a secret key $X \in \mathbb{Z}_q$ and the corresponding public key $Y = g^X \bmod p$. This key pair will be used for ElGamal threshold encryption.
2. The parties in P execute a second instance of the GJKR-DKG scheme to generate the value $h = Y^{\widehat{X}} \bmod p$.

II. KG phase (k-iterations):

1. *Generate random $x = f(0)$ for dealer P_D:*
   (a) Each party $P_i \in P$ chooses randomly $r_i \in_R \mathbb{Z}_q$ and broadcasts the commitment

   $$commit_i = g^{r_i} h^{r_i} \bmod p.$$

   If one party P_j doesn't broadcast her commitment she is disqualified. We denote by $QUAL \subseteq P$ the set of non-disqualified parties.
   (b) $P_i \in QUAL$ chooses randomly $z_i \in_R \mathbb{Z}_q$ and computes the ElGamal encryption of $F_i = F(z_i) \in G$:

   $$C_i = (D_i, E_i) = (g^{r_i}, Y^{r_i} \cdot F_i). \tag{2}$$

We set: $H_i = \frac{commit_i}{D_i} = h^{r_i}$. Besides, the following non-interactive zero-knowledge proof is generated:

$$NIZK_i = PoK\{r_i : H_i = h^{r_i} \wedge D_i = g^{r_i}\}.$$

P_i broadcasts C_i and $NIZK_i$ in $QUAL$. Obviously each party can easily verify the correctness of the zk-proof. Incorrect behavior leads to disqualification.
By using the multiplicative homomorphic property of the ElGamal encryption, each party can now compute the ciphertext of $F(\widetilde{z}) := \prod_{P_j \in QUAL} F(z_j)$, i.e.,

$$C = (\prod_{P_j \in QUAL} g^{r_j}, \prod_{P_j \in QUAL} Y^{r_j} F(z_j)). \tag{3}$$

(c) P_D chooses a uniformly distributed value $\widetilde{a}_0 \in \mathbb{Z}_q$ and broadcasts $\widetilde{A}_0 = g^{\widetilde{a}_0}$ in $QUAL$.

(d) P_D chooses a set T of $t+1$ parties for publishing their decryption shares of C. Hence, each party in P can decrypt the value C and finally knows the unique value $F_{all} = Decrypt(C)$. It follows that each party can easily compute the value $\widetilde{z} = F^{-1}(F_{all})$ by inverting F_{all}. Hence, P_i can calculate the unique verification value:

$$\begin{aligned} A_0 &= \widetilde{A}_0 \cdot g^{\widetilde{z}} \bmod p \\ &= g^{\widetilde{a}_0 + \widetilde{z}} \bmod p \\ &= g^{a_0} \bmod p. \end{aligned}$$

In particular, only P_D knows $a_0 = \widetilde{z} + \widetilde{a}_0$.

2. *Generate a polynomial to share the secret (Feldman-VSS):*

(a) P_D creates a random polynomial $\overline{f}(z)$ over $\mathbb{Z}_q$ with degree t:

$$\overline{f}(z) = a_1 z + \ldots + a_t z^t.$$

P_D chooses the value a_0 generated in II.1 as the constant coefficient. Hence, the polynomial to compute the parties' shares is composed of

$$f(z) = a_0 + \overline{f}(z) = a_0 + a_1 z + \ldots + a_t z^t.$$

P_D broadcasts the following verification values in $QUAL$:

$$A_k = g^{a_k}, \quad \text{with } k = 1, \ldots, t.$$

P_D computes $s_i = f(i) \bmod q$ and sends it securely to P_i.

(b) Each party verifies the shares she received from other participants by using the following equation:

$$g^{s_i} = \prod_{k=0}^{t} (A_k)^{i^k} \tag{4}$$

If the check fails P_i broadcasts a complaint against P_D.

(c) P_D can answer this accusal by publishing valid shares s_i, which satisfy equation 4.

(d) The dealer P_D is disqualified by each honest party $P_i \in QUAL$, if either:

- P_i receives more than t complaints, or
- P_D wasn't able to answer with valid shares s_i in II.2c.

The k-JRKG scheme is called *t-secure* if in the presence of an attacker that corrupts at most t parties, the following requirements for correctness and secrecy are satisfied:

DEFINITION 2.1 *The k-JRKG protocol is **t-correct**, if for all qualified dealers P_D with shared private key x and public key $y = g^x$ the following conditions hold:*

(C1) *All subsets of $t+1$ shares provided by honest parties define the same unique secret x.*

(C2) *All honest parties can compute the dealer's unique public key $y = g^x \bmod p$.*

(C3) *x is uniformly distributed in $\mathbb{Z}_q$ and hence y is uniformly distributed in the subgroup G.*

(C4) *Cheating leads to disqualification.*

Theorem 2.1 *For every polynomial-time bounded adversary which corrupts at most $t < n/2$ parties the following holds:*

If the DDH-assumption is true, then for each secret x distributed in the KG-phase the correctness properties of definition 2.1 hold.

The following theorem states that k-JRKG is t-secure:

Theorem 2.2 *For every polynomial-time bounded adversary which corrupts at most $t < \frac{n}{2}$ parties the following holds:*

If the DL-assumption is true, then for each secret x distributed in the KG-phase by an honest dealer P_D no information on x can be learned by the adversary except for what is implied by the publicly known value $y = g^x$.

References

[CS04] Canny, J., Sorkin, S.: Practical Large-Scale Distributed Key Generation. Lecture Notes in Computer Science **3027** (2004), 138–152

[GJ04] Golle, P., Juels, A.: Dining Cryptographers Revisited. Lecture Notes in Computer Science **3027** (2004), 456–473

[GJKR96] Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust Threshold DSS Signatures. Lecture Notes in Computer Science **1070** (1996), 354–371

[GJKR99] Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. Lecture Notes in Computer Science **1592** (1999), 295–310

[Ped91] Pedersen, T.: A threshold cryptosystem without a trusted party. Lecture Notes in Computer Science **547** (1991), 522–526

Theorie und Anwendungen von Tree Parity Machines für die Kryptographie

Andreas Ruttor* und Markus Volkmer†
*Universität Würzburg, Institut für Theoretische Physik und Astrophysik
Am Hubland, D-97074 Würzburg
†Technische Universität Hamburg Harburg, Institut für Rechnertechnologie
Schwarzenbergstraße 95, D-21073 Hamburg

Schlüsselaustauschprotokolle werden immer dann benötigt, wenn zwei Partner A und B einen geheimen Schlüssel vereinbaren möchten, aber nur einen öffentlichen Kommunikationskanal zur Verfügung haben. Zur Konstruktion eines solchen Algorithmus kann man auf einen Effekt zurückgreifen, der bei der Untersuchung neuronaler Netze gefunden wurde: zwei Tree Parity Machines (TPMs), die voneinander lernen, synchronisieren schneller als ein drittes Netzwerk, das nur passiv an der Kommunikation teilnimmt [1]. Ein erfolgreicher Angriff mit den bisher bekannten Methoden erfordert deshalb einen erheblich höheren Aufwand als der Schlüsselaustausch selbst.

Beide Partner verwenden für den neuronalen Schlüsselaustausch je eine aus K *hidden units* mit jeweils N Eingabeneuronen bestehende TPM, deren Anfangszustände zufällig und unabhängig voneinander gewählt werden. Jedes Netz definiert so eine Abbildung der KN binären Eingaben $x_{ij} \in \{-1, +1\}$ auf eine binäre Ausgabe $\tau \in \{-1, +1\}$,

$$\tau = \prod_{i=1}^{K} \sigma_i = \prod_{i=1}^{K} \operatorname{sgn}\left(\sum_{j=1}^{N} w_{ij} x_{ij} \right) ,$$

die durch ganzzahlige Gewichte $w_{ij} \in \{-L, -L+1, \ldots, L\}$ parametrisiert ist. Die TPMs von A und B erhalten in jedem Schritt einen zufällig erzeugten Satz der x_{ij} und lernen die Ausgabe τ ihres Partners. Dabei werden die Gewichte gemäß der Hebbschen Lernregel, $w_{ij}^{+} = w_{ij} + \tau\, x_{ij} \Theta(\sigma_i \tau) \Theta(\tau^A \tau^B)$, oder einer ihrer Varianten angepasst. Dies führt nach t_{sync} Schritten zu übereinstimmenden Gewichten ($w_{ij}^A = w_{ij}^B$), die anschließend von den Partnern als gemeinsamer Schlüssel eingesetzt werden können. Wegen der stochastischen Natur dieses Prozesses ist t_{sync} eine Zufallsvariable, deren Mittelwert für $K \leq 3$ proportional zu $L^2 \log N$ anwächst [2].

Sowohl bei der Synchronisation als auch beim Training einer weiteren TPM durch einen Angreifer E können zwei Arten von Lernschritten auftreten: attraktive und repulsive. Wenn die Ausgaben korrespondierender hidden units übereinstimmen ($\sigma_i^A = \sigma_i^B$), dann bewirkt die Lernregel gleiche Änderungen in beiden neuronalen Netzen. Wird dabei eines der Gewichte am Rand bei $-L$ oder $+L$ reflektiert, so verringert sich der Abstand $|w_{ij}^A - w_{ij}^B|$ durch diesen attraktiven Schritt. Repulsive Schritte treten dagegen bei abweichenden Ausgaben auf ($\sigma_i^A \neq \sigma_i^B$) und führen zu einer Desynchronisation der TPMs, weil nur in einer der beiden hidden units die Gewichte angepasst werden. Folglich bestimmen die Häufigkeiten attraktiver und repulsiver Schritte wesentlich die Geschwindigkeit der Synchronisation.

Da als Gesamtausgabe τ die Parität der σ_i verwendet wird, lassen sich repulsive Schritte an $\tau^A \neq \tau^B$ erkennen, wenn eine ungerade Anzahl von hidden units betroffen ist. Hier haben A und B einen für die Sicherheit des neuronalen Schlüsselaustauschs entscheidenden Vorteil. Sie können sich nämlich gegenseitig beeinflussen und überspringen auf diese Weise einen Teil der repulsiven Schritte. Der Angreifer dagegen hat diese Möglichkeit nicht und kommt deshalb bei der Synchronisation im

Mittel langsamer voran als die Partner. Auf Grund dieses Nachteils gelingt E nur mit geringer Wahrscheinlichkeit P_E eine Synchronisation mit A oder B, bevor diese einen gemeinsamen Schlüssel erzeugt haben und den Schlüsselaustausch beenden.

Die Erfolgswahrscheinlichkeit P_E wurde für viele verschiedene Angriffsmethoden [3] in Abhängigkeit von der synaptischen Tiefe L der TPMs untersucht. Dabei zeigt sich praktisch immer das gleiche Verhalten: P_E fällt exponentiell mit zunehmendem L ab, während t_{sync} nur proportional zu L^2 ansteigt [4, 5]. Die synaptische Tiefe hat also eine ähnliche Einfluss auf die Sicherheit des neuronalen Schlüsselaustauschs wie die Länge des Schlüssels bei einem Verschlüsselungsalgorithmus. Mit zunehmendem L steigt die Komplexität für einen erfolgreichen Angriff exponentiell an, während der Aufwand für die Erzeugung des gemeinsamen Schlüssels nur moderat wächst. Folglich kann man durch Erhöhen der synaptischen Tiefe L prinzipiell jedes gewünschte Sicherheitsniveau einstellen.

Aufgrund der einfachen Arithmetik auf vergleichsweise kleinen ganzen Zahlen können TPMs vorteilhaft in Hardware realisiert werden. Im Fachgebiet der sog. *Embedded Security* wird neben der effizienten Realisierung etablierter kryptographischer Verfahren auch an alternativen kryptographischen Verfahren geforscht. Man versucht Sicherheitslösungen auch für solche Systeme zu finden, die zum Teil extreme Beschränkungen hinsichtlich der zur Verfügung stehenden Rechenleistung, Stromverbrauch und oder Logik-Fläche aufweisen.

Im Rahmen des Projektes *Tree Parity Machine Rekeying Architectures* (TPMRA), an der TUHH von 2002-2006, wurde die Anwendung von TPMs in verschiedenen Systemen zur Absicherung von Kommunikation untersucht. Dabei wurden sowohl Software- als auch Hardware-Realisierungen [6] betrachtet. Im Unterschied zu den theoretisch erzielbaren Sicherheitsniveaus durch Skalierung der TPM-Parameter lag in diesem Projekt der Fokus auf möglichst hoher Sicherheit bei vergleichsweise kleinen Systemparametern, d.h. bei eher kleinen TPMs. Ein (bis dato) ausreichendes Sicherheitsniveau kann bei kleinen Systemen nur mit dem authentifizierten TPM-Schlüsselaustausch erzielt werden [8].

Neben der algorithmischen Variante mit Bit-Paket-Lernen wurden auch Varianten der Lernregel sowie Strategien des wiederholten Auffrischen von Sitzungsschlüsseln (sog. Rekeying-Strategien) mit dem Ziel der praktischen Anwendung untersucht. Aus dem Trajektorie-Modus der TPM wurde eine Stromchiffre abgeleitet, die sowohl OFB als auch CFB ermöglicht [7]. Die Eigenschaften und die Sicherheit der Chiffre befinden sich noch in Untersuchung.

Die voll-serielle (Hardware-)IP-Core-Realisierung einer TPMRA benötigt etwa 2400 Gatteräquivalente inklusive Speicher. Aufgrund dieser geringen Logik-Fläche eignet sie sich zum Einsatz auf ressourcenbeschränkten Geräten, wie z.B. RFID-Tags [7]. Eine weitere Anwendung ist die Einbettung der (semi-parallelen) TPMRA in Bus-Systeme. Hierbei kommt sowohl die geringe Fläche (als kostengünstige Hardware-Komponente) als auch die Multifunktionalität des Verfahrens zugute. Identifikation, Schlüsselaustausch und Verschlüsselung per Stromchiffre können aus einem Prinzip abgeleitet werden [9]. Die Verwendung einer voll-parallelen TPMRA im Trajektorie-Modus erlaubte es, in einem PCI-Bus-System für jeden PCI-Buszugriff einen neuen Schlüssel bereitzustellen.

TPMs erlauben es zudem einen Mehrparteien-Schlüsselaustausch über denselben Synchronisationseffekt zu realisieren. Diese Variante wurde prototypisch in einem WLAN-AdHoc Netzwerk zur Absicherung von Gruppenkommunikation auf Laptops implementiert. Hierbei können sowohl parallele als auch sequentielle Synchronisationsprozesse genutzt werden.

Literatur

[1] Ido Kanter, Wolfgang Kinzel, and Eran Kanter. Secure exchange of information by synchronization of neural networks. *Europhys. Lett.*, 57(1):141–147, January 2002.

[2] Andreas Ruttor, Georg Reents, and Wolfgang Kinzel. Synchronization of random walks with reflecting boundaries. *J. Phys. A*, 37:8609–8618, August 2004.

[3] Alexander Klimov, Anton Mityaguine, and Adi Shamir. Analysis of Neural Cryptography. In Yuliang Zheng, editor, *Advances in Cryptology—ASIACRYPT 2002*, page 288, Heidelberg, February 2003. Springer.

[4] Rachel Mislovaty, Einat Klein, Ido Kanter, and Wolfgang Kinzel. Public channel cryptography by synchronization of neural networks and chaotic maps. *Phys. Rev. Lett.*, 91(11):118701, 2003.

[5] Andreas Ruttor, Wolfgang Kinzel, Rivka Naeh, and Ido Kanter. Genetic attack on neural cryptography. *Phys. Rev. E*, 73(3):036121, March 2006.

[6] Markus Volkmer and Sebastian Wallner. Tree Parity Machine Rekeying Architectures. *IEEE Transactions on Computers*, 54(4) 2005.

[7] Markus Volkmer and Sebastian Wallner. Lightweight Key Exchange and Stream Cipher based solely on Tree Parity Machines. ECRYPT Workshop on RFID and Lightweight Crypto, 2005, Graz University of Technology, Graz, Austria, IACR Cryptology ePrint Archive, Report 2005/232.

[8] Markus Volkmer. Entity Authentication and Authenticated Key Exchange with Tree Parity Machines. IACR Cryptology ePrint Archive, Report 2006/112, March 2006.

[9] Sascha Mühlbach, Markus Volkmer, and Sebastian Wallner. Encrypted and Authenticated Communication via Tree-Parity Machines in AMBA Bus Systems. 4. Krypto-Tag – Workshop über Kryptographie 2006, Ruhr-Universität Bochum, Horst-Görtz-Institut für IT-Sicherheit, Technical Report No. NDS-1/06, Ulrich Greveler (Hrsg.).

Opportunistische E-Mail-Sicherheit

Alexander Naumann* und Tobias Straub†

* Technische Universität Darmstadt, E-Mail: alexander.naumann@web.de

† Fraunhofer-Institut für Sichere Informationstechnologie SIT, Darmstadt

1 Einleitung

Die meisten der aktuellen E-Mail-Programme unterstützen sichere E-Mail mittels S/MIME oder PGP. Allerdings setzen Benutzer diese Funktionen aufgrund der aus ihrer Sicht ungünstigen Kosten-Nutzen-Relation sehr selten in der Praxis ein [1].

Ein Grund dafür ist die hohe technische Komplexität sowie das für ein korrektes Funktionieren der Sicherheitsmechanismen erforderliche Fachwissen. Insbesondere für Technik-Laien ist der Aufwand für die Konfiguration eines E-Mail-Accounts mit Zertifikaten und für die Benutzung im täglichen Einsatz zu hoch.

In dieser Arbeit wird ein opportunistischer Ansatz vorgestellt, der durch neue Interaktionsmuster und Heuristiken den Aufwand für den Benutzer senken, gleichzeitig aber ein möglichst hohes Maß an Sicherheit für E-Mails gewährleisten soll. Als Ziel wird ein sinnvoller und praxisnaher Ausgleich zwischen Benutzbarkeit und Sicherheit angestrebt.

Im nächsten Abschnitt stellen wir unseren Ansatz von opportunistischer E-Mail-Sicherheit vor, wie er bereits in Teilen prototypisch umgesetzt wurde (siehe Abschnitt 3). In Abschnitt 4 gehen wir auf Erweiterungen ein, die eine Integration in das Nutzer-Interface von Mozilla Thunderbird vorsehen.

2 Lösungsansatz

Unser Lösungsansatz lässt sich von folgenden Kriterien leiten:

- Sicherheitskritische Routine- und Management-Aufgaben werden – soweit wie möglich – vom E-Mail-Programm übernommen.
- Fehlerfälle oder mögliche Angriffe und entsprechende Reaktionen werden bereits frühzeitig beim Systemdesign berücksichtigt.
- Die notwendigen Vorkenntnisse werden als minimal angenommen, so dass auch Laien hinreichend verlässlich sichere E-Mail verwenden können.
- Der Benutzer wird nicht mit technischen Details konfrontiert, sofern er dies nicht ausdrücklich wünscht.

Die ersten beiden Kriterien führen zu einer Reihe von Heuristiken für die Anwendung von Security Policies sowie die Behandlung von Fehlern und Signalisierung möglicher Angriffe. Aufgrund des dritten und vierten Kriteriums sehen wir ein Konzept verschiedener „User Levels“, d.h. Kenntnisstufen vor, die den Grad an jeweils vorhandener Transparenz und Automatisierung bestimmen (vgl. [2]).

Typischerweise erfordern heutige E-Mail-Programme häufige Benutzerinteraktionen für sicherheitsrelevante Einstellungen und Vorgänge. Demgegenüber besteht die Idee opportunistischer Sicherheit in einem *Best Effort*-Prinzip, d.h. E-Mails werden immer automatisch und ohne Benutzerinteraktion verschlüsselt und signiert, sobald sich dazu eine Gelegenheit ergibt. Signiert werden E-Mails, sofern bekannt ist, dass der Kommunikationspartner das System ebenfalls nutzt, was sich aus vorherigen E-Mails schließen lässt. Verschlüsselt wird, sofern ein Schlüssels des Kommunikationspartners bekannt ist, ansonsten erfolgt der Versand unverschlüsselt. Eine weitere Eigenheit opportunistischer Sicherheit betrifft die Gültigkeitsprüfung von Zertifikaten. Im Gegensatz zu sonst gängiger Praxis ist die Verifikation eine Option bzw. es wird ganz auf Zertifikate verzichtet. Die genannten Prinzipien sind eine Verallgemeinerung des Konzepts der *opportunistic encryption*, wie sie etwa auch schon für sichere E-Mail eingesetzt wird [3].

Weiter werden der Sicherheitsstatus der Kommunikationspartner überwacht und Auffälligkeiten behandelt. Dazu gehört zum Beispiel eine nicht signiert empfangene E-Mail, obwohl der Sender dazu eigentlich technisch in der Lage sein sollte. Veränderungen bei der Sicherheit des E-Mail-Verkehrs oder Angaben zum Sicherheitsstatus einer E-Mail werden dem Nutzer deutlich gemacht.

3 Prototyp

Im Rahmen eines Praktikums an der TU Darmstadt wurde prototypisch ein „Opportunistic E-Mail Security System“ (OESS) in Form eines eigenständigen POP3-/SMTP-Proxy entwickelt. Da hierbei die Interaktion mit dem Benutzer über das E-Mail-Programm stark eingeschränkt ist, soll das System um eine graphische Benutzerschnittstelle erweitert werden, über die der Sicherheitsstatus und eventuelle Probleme besser wiedergegeben werden können.

Der in Java geschriebene Prototyp übernimmt die kryptographischen Operationen transparent für den Benutzer und dessen E-Mail-Programm. Der Proxy erhält die E-Mails vom Mail User Agent, bearbeitet diese in der oben beschriebenen Weise und leitet sie, mit nur geringer Verzögerung, an den Mail Transfer Agent weiter. Das E-Mail-Programm ist dazu für jeden vom Proxy zu verwaltenden Account einmalig so zu konfigurieren, dass der Datenverkehr über den Proxy geleitet wird.

Vom E-Mail-Programm wird eine Nachricht zum Versand vom Proxy in eine neue E-Mail eingebettet, deren Header nur die Felder `From`, `To` und `Subject` umfasst. Die Authentizität und Integrität dieser Header wird durch eine Signatur geschützt. Außerdem ist die Vertraulichkeit der Betreffzeile dadurch gegeben, dass das Programm in der „äußeren“ E-Mail nur einen Hinweis auf die Verwendung von OESS enthält und die ursprüngliche Betreffzeile mit der „inneren“ E-Mail verschlüsselt übertragen wird. Diese Sicherheitseigenschaften der E-Mail-Header bieten Ansätzen wie S/MIME und PGP nicht.

Der Prototyp nutzt derzeit ein proprietäres Datenformat, wobei jedoch noch geprüft wird, ob dieses nicht auch kompatibel zu S/MIME oder PGP gemacht werden kann (vgl. [4]). Eine Idee ist dabei, die komplette gesicherte E-Mail in ein S/MIME-kompatibles Format zu verpacken und zu verschicken.

Gemäß kryptographischer *Best Practice* wird zuerst signiert, anschließend gegebenenfalls verschlüsselt. Zum Key Management haben wir einen Mechanismus entwickelt, der den Austausch der Schlüssel regelt, für Kontinuität beim Key-Rollover sorgt und somit durchgängige Sicherheit leistet.

Bei gängigen E-Mail-Programmen kann eine E-Mail nur dann verschlüsselt an mehrere Empfänger verschickt werden, wenn für jeden dieser Empfänger ein Zertifikat vorliegt. Andernfalls ist die Op-

tion Verschlüsselung deaktiviert und die Nachricht wird an alle unverschlüsselt versendet. Dagegen ist der Proxy in der Lage, eine E-Mail an verschiedene Empfänger so zu splitten, dass Empfänger, die OESS-fähig sind, ihre E-Mail geschützt und alle anderen eine ungeschützte Version erhalten. Durch den *Best Effort*-Ansatz ist somit unter dem Strich ein höherer Anteil an verschlüsselten Nachrichten erreicht worden.

4 User Interface-Integration

Zwar lässt sich der Proxy auch stand-alone betreiben, jedoch entfällt dabei die Möglichkeit, dem Benutzer Feedback über die Sicherheit seines E-Mail-Verkehrs zu geben. Außerdem lassen sich keine qualitativen oder quantitativen Aussagen über den E-Mail-Verkehr mit einem Kommunikationspartner machen, etwa in Bezug auf die bisherige Absicherung der E-Mails oder den Einsatz anderer E-Mail-Verschlüsselungssoftware.

Ziel der Visualisierung ist es nun, dem Nutzer wichtige Informationen über die E-Mail darzustellen und gegebenenfalls Meldungen auszugeben oder Interaktionen zu ermöglichen. Der Nutzer muss wissen, welche der möglichen Aktionen potenziell gefährlich sind, wie sie zu vermeiden sind und wie sein aktueller Sicherheitstatus insgesamt und der einer einzelnen E-Mail ist. Dafür wird als Machbarkeitsstudie eine Extension für Mozilla Thunderbird [5] implementiert. Dieses Visualisierungs-Plugin soll demonstrieren, wie entsprechende Informationen sinnvoll dargestellt werden können. Je nach User Level werden mehr oder weniger Informationen angezeigt, bzw. finden mehr oder weniger Interaktionen statt:

	User Level		
Visualisierung	Anfänger	Fortgeschrittener	Experte
Status der Mail (signiert, verschlüsselt)	ja	ja	ja
OESS-fähige Partner	ja	ja	ja
Zeitliche Entwicklung des Sicherheitsstatus	nein	ja	ja
Fehlermeldungen	nein	ja	ja
Abfrage bei E-Mail an mehrere Empfänger	nein	ja	ja
Sender verwendet abgelaufene Schlüssel	nein	nein	ja

Die Evaluation des Systems soll unter möglichst realitätsnahen Bedingungen stattfinden, wobei zunächst untersucht wird, welche Usability-Methoden sich dazu am besten eignen. Die Evaluation selbst misst die Größen Nutzer-Akzeptanz (durch Interviews oder Umfragen), Effektivität der Sicherheitsmechanismen (durch Angriffsszenarien und Erkennung des Angriffe) sowie den Zeitaufwand für die Installation und Bedienung.

5 Ausblick

Denkbar ist, das Prinzip opportunistischer Sicherheit auch auf andere Anwendungsbereiche von PKI zu übertragen. Wo und in wieweit dies möglich ist, soll untersucht werden. Einsatzgebiete könnten beispielsweise File Sharing- und Chat-Anwendungen oder die Absicherung von Datenverbindungen über HTTPS sein. Ähnliche Ansätze werden etwa bereits bei SSH oder IPv6 [6] verfolgt.

Literatur

[1] Tobias Straub. Usability Challenges of PKI. Dissertation TU Darmstadt, 2005. http://elib.tu-darmstadt.de/diss/000682

[2] A. Whitten und J.D. Tygar. Safe Staging for Computer Security. In Proc. of the Workshop on Human-Computer Interaction and Security Systems, 2003

[3] Simson L. Garfinkel. Enabling email confidentiality through the use of opportunistic encryption. In Proc. National Conference on Digital Government Research, 2003

[4] Simson L. Garfinkel, Robert C. Miller. Johnny 2: a user test of key continuity management with S/MIME and Outlook Express. In Proc. of the 2005 Symposium on usable privacy and security, 2005

[5] Mozilla Thunderbird Homepage. http://www.mozilla.org/products/thunderbird (Abgerufen am 20.05.2006)

[6] Claude Castelluccia, Gabriel Montenegro, Julien Laganier, Christoph Neumann. Hindering Eavesdropping via IPv6 Opportunistic Encryption. In ESORICS, pages 309-321, 2004

Sicherheitsbeweise für zertifikatlose Public-Key Schemata

Ewan Fleischmann

Universität Mannheim
Fakultät für Mathematik und Informatik
Lehrstuhl für Theoretische Informatik

Das Konzept der zertifikatlosen Public-Key Kryptographie (PKC) wurde im Jahre 2003 auf der AsiaCrypt erstmalig von Al-Riyami vorgestellt [AlR]. Einzuordnen ist sie zwischen der klassischen zertifikatbasierten PKC und der identitätsbasierten PKC. Ziel war es, von den Vorteilen der identitätsbasierten PKC zu profitieren (bspw. keine Public-Key Infrastruktur nötig, komplexer kryptographischer Workflow möglich) ohne jedoch deren oftmals inakzeptable Nachteile (zwangsweise Schlüsselhinterlegung) in Kauf nehmen zu müssen. Technisch basiert die zertifikatlose PKC jedoch weiterhin auf einer vertrauenswürdigen dritten Partei (genannt Key Generation Center, KGC). Diese kann jedoch keine Chiffretexte entschlüsseln, wie dies im identitätsbasierten Fall möglich ist.

Bei einem Sicherheitsbeweis muss hierbei von zwei möglichen Angreifern ausgegangen werden: Ein Angreifer (Typ-I) entspricht einem „normalen“ Angreifer einer beliebigen dritten Partei (ähnlich dem eines Public-Key Schemas). Der zweite Angreifer (Typ-II) modelliert ein KGC, welches versucht Zugriff auf den Klartext einer verschlüsselten Nachricht zu bekommen. Dieses hat jedoch Zusatzinformationen, über welche ein Typ-I Angreifer nicht verfügt. Damit ein Schema sicher ist, muss es sowohl sicher sein gegenüber einem Angreifer vom Typ-I wie gegenüber einem Angreifer vom Typ-II.

Bis jetzt wurden nur zertifikatlose Schemata vorgestellt, welche entweder in einem eingeschränkten Angriffsmodell oder im Zufallsorakelmodell sicher sind. Als das zentrale Problem bei der Beweisführung im Standardmodell mit der üblichen (reduktionistischen) Vorgehensweise stellt sich heraus, dass ein Schema, welches beweisbar sicher gegenüber einem Typ-I Angreifer ist, zwangsläufig unsicher gegenüber einem Typ-II Angreifer ist. Aufgrund dieses sehr harten Resultats gingen einige Kryptographen davon aus, dass es prinzipiell nicht möglich sein kann, die Sicherheit von zertifikatlosen Schemata im Standardmodell (ohne weitere Einschränkungen vorzunehmen) zu beweisen. Dass dies nicht zwangsläufig daraus folgt, wird in [DK] genauer untersucht. Auch werden mögliche Lösungsansätze aufgezeigt.

Inhalt des Vortrages ist eine kurze Darstellung der zentralen Ideen der identitätsbasierten und zertifikatlosen PKC. Dabei wird auch ein vereinfachtes Angriffsmodell auf diese Schemata vorgestellt, welches im Rahmen der Diplomarbeit entwickelt wurde. Daran anschließend wird die Problematik der Beweisführung im Standardmodell bei zertifikatlosen Schemata erläutert und auf einige konkrete Lösungsansätze eingegangen.

Literatur

[AlR] Sattam S. Al-Riyami. Cryptographic Schemes based on Elliptic Curve Pairings. PhD Thesis, 2004, University of London, Department of Mathematics

[DK] Alexander Dent and Caroline Kudla. On Proofs of Security for Certificatelesss Cryptosystems, Information Security Group, Royal Holloway, Cryptology ePrint Archive, Report 2005/348

Universelle Message Authentication Codes

Christian Forler

HORNBACH Baumarkt AG, Hornbachstraße, 76878 Bornheim

In den letzten Jahren wurden Schwächen in den gängigen kryptographischen Hashfunktionen MD5 oder SHA-1 gefunden. Standardisierte Authentifikationsverfahren wie DSS/DSA oder HMAC basieren auf diesen Hashfunktionen. Dieser Umstand macht diese Verfahren unsicher, da sich aus unsicheren Bausteinen kein sicheres Authentifikationsverfahren konstruieren lässt. Mikle beschreibt beispielsweise, wie Authentifikationsverfahren, die auf MD5 basieren, gebrochen werden können [MIK04].
Authentifikationsverfahren wie der XOR-MAC, die auf Blockchiffren beruhen, sind sicher, wenn die verwendete Blockchiffre sicher ist. Leider gibt es noch keinen Sicherheitsbeweis für moderne Blockchiffren wie AES. Weiterhin sind solche Authentifikationsverfahren signifikant langsamer als solche, die auf kryptographischen Hashfunktionen basieren. Dies liegt daran, dass kryptographische Hashfunktionen in der Regel schneller als Blockchiffren sind.
Wegman und Carter haben 1981 gezeigt, dass sich (informationstheoretisch) sichere MACs auf Basis von Hashfunktionen, die paarweise kollisionsresistent sind, konstruieren lassen [CW81]. Diese Hashfunktionen lassen sich nicht nur für die Kryptographie sondern auch für andere Bereiche der Informatik (Datenbanken, Filesysteme,...) nutzen. Daher werden sie universelle Hashfunktionen genannt. MACs, die darauf basieren, sind nicht nur beweisbar sicher sondern auch hoch performant. Wie Performancemessungen zeigen, lassen sich auf Standardhardware damit Durchsatzraten von unter einem cpb (clock cycle per byte) erreichen [KRO00].
Es spricht daher nichts dagegen, sich intensiver mit solchen universellen MACs zu beschäftigen.

Literatur

[MIK04] Ondrej Mikle. Practical Attacks on Digital Signatures Using MD5 Message Digest. December 2004.

[CW81] J. Lawrence Carter and Mark N. Wegman. New Hash Functions and Their Use in Authentication and Set Equality. Journal of Computer and System Sciences, 1981.

[KRO00] Theodore D. Krovetz. Software-Optimized Universal Hashing and Message Authentication. September 2000.

Privacy Friendly Location Based Service Protocols using Efficient Oblivious Transfer

Markulf Kohlweiss* and Bartek Gedrojc†

* KU-Leuven
Leuven
Belgium

† TU-Delft
Delft
Nederlands

Mobile devices add an additional dimension to context-based services: location. Bob, providing a *location-based service* (LBS), uses the location of Alice to answer her request, e.g., to find the next Italian restaurant. From a security standpoint the two main assets to be protected are Bob's database, and the location of Alice. Cryptographically this problem corresponds to an *oblivious transfer* (OT) of Bob's location specific data, where the index of the 1-out-of-n OT is the location σ of Alice, $1 \leq \sigma \leq n$. By the properties of the OT, Alice learns only the information of the single map cell σ, while Bob is oblivious of Alice's location.

In our work we investigate the specific needs of privacy friendly LBSs, and we design solutions based on efficient OT that take them into considerations. For instance, service providers have an interest in reducing the costs of the OT through economies of scale. Adaptive OT, where the same database is queried with little additional cost, provides a natural starting point. Similarly, the restricted capabilities of mobile users require a careful design of the system. In many of today's mobile networks there exists a dedicated party, the operator, that knows Alice's location. We investigate the role of this party to act as a proxy that inputs the user's location to the protocol and helps with doing the computation, but which otherwise remains oblivious of the protocols result.

Finally we investigate the use of homomorphic encryption in order to support the access of multiple LBSs by the same user. This can be seen as a split oblivious transfer involving up to ℓ location-based services simultaneously. Each LBS is handling a different database. Again the privacy sensitive information, i.e., which services Alice subscribed to, remains hidden from everyone else. Moreover the homomorphic property is utilized to facilitate the privacy preserving payment of the services.

Privacy friendly LBS. Privacy is an enormous topic [12]. It is a sociological phenomenon which has many legal and commercial implications. The different ways location is used by an LBS greatly influences Alice's privacy experience. Is she interacting only with the service or with other users of the service? Is her location only used at the time of her request, or is she constantly tracked and notified upon certain events? Rather than covering all of this topics, we focus on a very specific sub-problem. Some of the techniques employed can however also be use for improving the privacy properties of other types of LBS protocols as surveyed in [11].

We do not consider solutions involving anonymity or service-side location specific privacy policies. Moreover we consider only solutions where no information at all about Alice's location is revealed to the LBS. The goal is to base the security of the system only on information theory and complexity theoretic assumptions. After the execution of the protocol a malicious LBS (even if collaborating with the operator) cannot compute anything, he (they) could not have computed before. It is easy to see that in this setting a notification service that contacts the user only upon events is impossible. The knowledge that the event occurred would already reveal information about Alice's location to the LBS.

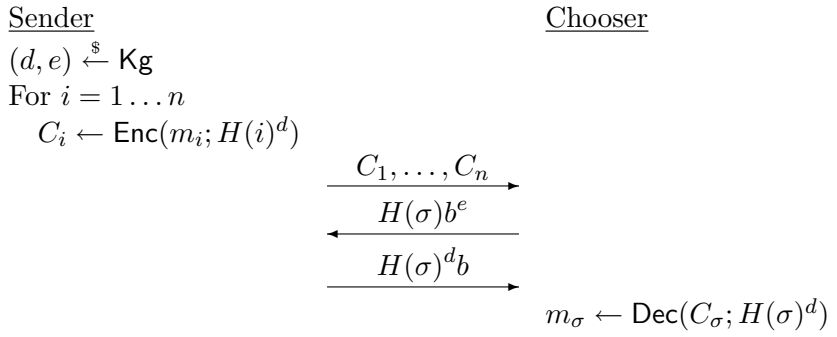

Figure 2: Adaptive OT based on Chaum blind signatures

Oblivious transfer. We model a privacy friendly LBS as a database that maps every location i to some information m_i. The number of different locations is restricted to n. A location can for instance be the name of a region, or a cell of a certain size on a map. Now the provisioning of a service corresponds to the retrieval of m_σ for a hidden σ from a database $m_1, \ldots, m_n$. We call σ also the index into the database. The privacy requirements of the user imply the need for private information retrieval (PIR) [5]. Symmetric PIR (SPIR) is required if the LBS wants to avoid leakage of information about locations that have not been queried. It was shown that for the case where there is only one copy of the database there exists a communication-efficient reduction from any PIR protocol to a 1-out-of-n OT. Moreover for the single copy case SPIR corresponds to 1-out-of-n OT (OT_n^1) [7].

Oblivious transfer was first introduced by Rabin [18]. It captures the on first sight paradoxical notion of a protocol by which a sender sends some information to the receiver, but remains oblivious as to what is sent. The paradox is resolved by recognizing that it are the actions of the *receiver and the sender* that determine the outcome of the protocol. Even [8] generalized it to 1-out-of-2 oblivious transfer (OT_2^1). The receiver determines which message out of two possible messages she is going to receive. In turn it was shown how to construct OT_n^1 from n [2] and even $\log n$ [13] applications of OT_2^1. [15, 1, 10] provided direct constructions for OT_n^1 based on the decisional Diffie-Hellman and quadratic residuosity assumptions.

Adaptive OT For location-based services, we are not so much interested in single executions of oblivious transfer, but want to query the same database multiple times at different indexes. This can be achieved by letting the sender commit to the database and running OT_n^1 multiple times. However this is not the most efficient solution. Moreover the security requirements of such a system differ from those of normal oblivious transfer, as the protocol keeps internal state and queries can be chosen adaptively based on the results of previous queries. The first adaptive oblivious transfer protocol was proposed in [14]. Recently more efficient schemes were proposed by [16, 6]. [4] recognized that the last two schemes are based on a common principle to construct adaptive oblivious transfer from unique blind signature schemes.

We briefly sketch the basic idea of the scheme using an example based on Chaum blind signatures (cf. Fig. 2). First, all messages are symmetrically encrypted using the RSA signature of the index. $H(\cdot)$ is a full domain cryptographic hash function. The encrypted database is transferred to Alice. When Alice wants to obtain the information for location σ, she runs a Chaum blind signature protocol with the sender to obtain the key.

Dynamic OT For practicality reasons we are also interested in dynamic databases that can shrink and grow during the execution of the adaptive oblivious transfer. This allows us to update parts of the database. For an update a new message is added to the database. Instead of accessing an old index, the user now has to access the new index. We require an additional table, that maps locations to their current indices. This update procedure reveals information about the database as Alice learns which entries have changed. It is an open research question whether we can do updates which don't reveal any information but are still substantially more efficient than running the whole protocol with a new database.

Increasing the size of the database is straightforward. The sender just transfers a new cipher text $C_{n+1} = \mathsf{Enc}(m_{n+1}; H(n+1)^d)$ and transfers it to the sender. The sender can now also ask for blind signatures on $n+1$. And decrypt C_{n+1}.

Deletion is more complicated. Our approach is to let the receiver prove that the requested σ is in the set of still valid indices V, *e.g.*, by using a dynamic accumulator [3]. Together with every C_i the sender now sends a witness $w_i = v^{(p_i^{-1})}$, with p_i a prime. Before obtaining the signature the receiver now needs to prove that she knows a valid witness that corresponds to the blind signature request for index σ. For efficient protocols, it is now no longer possible to use a full domain hash RSA signature.

Proxy OT and multi-database extensions For today's mobile networks it is natural to assume a third party, the operator, that knows Alice's location and can help her in executing the OT despite of limited device capabilities. This party executes most of the receiver's part of the OT protocol, but only Alice obtains the final result that allows her to decrypt C_σ. We call the third party a proxy and the new protocol a proxy OT protocol.

In location-based services, not only Alice's location, but also the type of service she is accessing is privacy sensitive information, which we do not want to reveal to the operator, or even the service himself. Thus a solution to this problem is of particular importance for LBS which use proxy OT, but may be of individual interest as an independent primitive. The selection of up to k services can be interpreted as an additional k-out-of-ℓ OT, which is run independently but interleaving the proxy OT. Only Alice knows which of the ℓ services she is accessing. Additional extensions are needed to facilitate payment for such hidden service usage. Preliminary ideas for a comprehensive solution are based on the use of homomorphic encryption in PIR [17], payment [1], and voting schemes [9].

References

[1] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 119–135, Innsbruck, Austria, May 6–10, 2001. Springer-Verlag, Berlin, Germany.

[2] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 234–238, Santa Barbara, CA, USA, August 1987. Springer-Verlag, Berlin, Germany.

[3] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2002.

[4] Jan Camenisch, Gregory Neven, and abi shelat. Adaptive oblivious transfer from blind signatures. unpublished manuscript through personal communication, 2006.

[5] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[6] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In Serge Vaudenay, editor, *PKC 2005*, volume 3386 of *LNCS*, pages 172–183, Les Diablerets, Switzerland, January 23–26, 2005. Springer-Verlag, Berlin, Germany.

[7] Giovanni Di Crescenzo, Tal Malkin, and Rafail Ostrovsky. Single database private information retrieval implies oblivious transfer. In *EUROCRYPT*, pages 122–138, 2000.

[8] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the Association for Computing Machinery*, 28(6):637–647, 1985.

[9] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In *EUROCRYPT*, pages 539–556, 2000.

[10] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 78–95, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany.

[11] Tobias Kölsch, Lothar Fritsch, Markulf Kohlweiss, and Dogan Kesdogan. Privacy for profitable location based services. In Dieter Hutter and Markus Ullmann, editors, *SPC*, volume 3450 of *Lecture Notes in Computer Science*, pages 164–178. Springer, 2005.

[12] Scott Lederer, Jennifer Mankoff, and Anind K. Dey. Towards a deconstruction of the privacy space. Featured at Ubicomp 2003, `http://guir.berkeley.edu/pubs/ubicomp2003/privacyspace.pdf`.

[13] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *31st ACM STOC*, pages 245–254, Atlanta, Georgia, USA, May 1–4, 1999. ACM Press.

[14] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 573–590, Santa Barbara, CA, USA, August 15–19, 1999. Springer-Verlag, Berlin, Germany.

[15] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *12th SODA*, pages 448–457, Washington, DC, USA, January 7–9, 2001. ACM-SIAM.

[16] Wakaha Ogata and Kaoru Kurosawa. Oblivious keyword search. *J. Complexity* , 20(2-3):356–371, 2004.

[17] Rafail Ostrovsky and William E. Skeith III. Private searching on streaming data. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 223–240. Springer, 2005.

[18] Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

Google Reveals Cryptographic Secrets

Emin Islam Tatlı

Department of Computer Science, University of Mannheim
tatli@th.informatik.uni-mannheim.de
http://th.informatik.uni-mannheim.de/people/tatli.shtml

Google hacking is a term to describe the search queries that find out security and privacy flaws. Finding vulnerable servers and web applications, server fingerprinting, accessing to admin and user login pages and revealing username-passwords are all possible in Google with a single click. Google can also reveal secrets of cryptography applications, i.e., *clear text and hashed passwords, secret and private keys, encrypted messages, signed messages* etc. In this paper, advanced search techniques in Google and the search queries that reveal cryptographic secrets are explained with examples in details.

1 Motivation

Having an index with over 25 billion entries, Google is the most popular web search engine. It indexes any information from web servers thanks to its hardworking web crawlers. But many sensitive data that should be kept secret and confidential are indexed by Google, too. Vulnerable servers and web applications, username-passwords for login sites, admin interfaces of database servers and online devices like web cameras without any access control, reports of security scanners and many more private information are available to hackers via Google.

This paper focuses on the advanced search queries that enable users to search different cryptographic values which are expected to stay private and safe. The paper is organized as follows: Section 2 summarizes the useful parameters for the advanced search in Google. In Section 3, examples of search queries for each type of cryptographic secret are illustrated. Finally, Section 4 explains possible security measures against Google hacking.

2 Advanced Parameters

Google supports many parameters for the advanced search and filters its results according to the parameters given by the user.

The *[all]inurl* parameter is used to filter out the results according to if the url contains a certain keyword or not. If more keywords are needed, the *allinurl* parameter should be used. *[all]intitle* filters the results according to the title of web pages. *[all]intext* searches keywords in the body of web pages. With the parameter *site* you can do host-specific search. *filetype* and *ext* parameters have the same functionality and are needed to filter out the results based on the file extensions like html, php, asp etc. The minus sign (-) can be put before any advanced parameter and reverses its behavior. As an example, a search containing the parameter *-site:www.example.com* will not list the results from www.example.com. The sign `"|"` stands for the logical OR operation.

3 Google Search for Cryptographic Values

From the cryptographic perspective, Google reveals also cryptographic secrets. Google can find out hashed passwords, secret keys, public and private keys, encrypted and signed files. What you need to do is only to enter the relevant search terms as explained in the following sections and click the search button.

3.1 Hashed Passwords

Database structures and contents can be backed up in *dump* files. The following query searches for SQL clauses that may contain usernames and passwords in cleartext or in hashed values within dump files. Hash and encryption relevant keywords can also be searched within files.

```
"create table" "insert into""pass|passwd|password"(ext:sql | | ext:dump | ext:dmp)
```

```
intext:"password|pass|passwd" intext:"md5|sha1|crypt" (ext:sql | ext:dump | ext:dmp)
```

3.2 Secret Keys

Since the secret keys are generated mostly as session keys and destroyed after the session is closed, they are not stored on disks permanently. But there are still some applications that need to store secret keys , e.g., Kerberos [9] shares a secret key with each registered principal for authentication purposes.

The following query lists the configuration files of a key distribution center (KDC) in Kerberos. Within the configuration files, the path of principal databases which contain principal ids and their secret keys is specified.

```
inurl:"kdc.conf" ext:conf
```

To find dumped Kerberos principal databases:

```
inurl:"slave_datatrans" OR inurl:"from_master"
```

Java provides a tool named *keytool* to create and manage secret keys in keystores. The extension of such keystores is *ks*. The following query searches for java keystores that may contain secret keys. Note that keytool can also manage private keys and certificate chains.

```
keystore ext:ks
```

3.3 Public Keys

Public keys, as the name implies, are public information and not secret. But for the sake of completeness, the search queries that list public keys are also written in this section.

To list PGP public key files:

```
"BEGIN PGP PUBLIC KEY BLOCK" (ext:txt | ext:asc | ext:key)
```

To list public keys in certificate files:

```
"Certificate:Data:Version" "BEGIN CERTIFICATE" (ext:crt | ext:asc | ext:txt)
```

3.4 Private Keys

Private keys should be kept *secret* for personal use but the following search queries show that people do not care about it and make it publicly accessible.

```
"BEGIN (DSA|RSA)" ext:key
```

```
"BEGIN PGP PRIVATE KEY BLOCK" inurl:txt|asc
```

Gnupg [5] encodes the private key in *secring.gpg*. The following search reveals secring.gpg files:

```
"index of" "secring.gpg"
```

3.5 Encrypted Files

For confidentiality, cryptography provides encryption of data. By encrypting, one can store sensitive files and emails securely on local storage devices. The following queries search for encrypted files and emails. It is sure that you need to know the relevant keys to decrypt but as shown in the previous examples, it is also possible to find secret keys and private keys. Besides, other crypto analysis techniques can help to decrypt the encrypted files.

The files that are encrypted with Gnupg get the extension *gpg* for binary encoding and the extension *asc* for ASCII encoding. The following first query searches files with gpg extension and tries to eliminate signed and public key files from the results. The second query lists ASCII encoded encrypted files. But note that signed files have also the same pattern and can be returned with the second query:

```
-"public|pubring|pubkey|signature|pgp|and|or|release" ext:gpg
```

```
-"BEGIN PGP MESSAGE" ext:asc
```

Many encryption applications use the extension *enc* for the encrypted files. There are some exceptions like AxCrypt File Encryption Software [6] which uses the extension *axx* for encrypted files:

```
-intext:"and" (ext:enc | ext:axx)
```

In XML Security, the encrypted parts of messages are encoded under *CipherValue* element:

```
"ciphervalue" ext:xml
```

3.6 Signed Messages

Digital signatures provide integrity, authenticity and non-repudiation in cryptography. The following searches list some signed messages, signed emails and file signatures.

To list pgp signed messages (*emails excluded*):

```
"BEGIN PGP SIGNED MESSAGE" -"From" (ext:txt | ext:asc | ext:xml)
```

To list signed emails:

```
"BEGIN PGP SIGNED MESSAGE" "From" "Date" "Subject" (ext:eml | ext:txt | ext:asc)
```

To list file signatures:

```
-"and|or" "BEGIN PGP SIGNATURE" ext:asc
```

4 Countermeasures

Google hacking can be very harmful and therefore the required security measures should be taken against it. One method is using automatic scan tools [2, 3, 4] that search possible Google hacks for a given host. You can use the tools to search for the available flaws and risks in your system. The tools mostly use the hack database [1] when they do scan. Another solution is integration of robots.txt (robots exclusion standard) [7] files in your system. Web crawlers (*hopefully*) respect the directives specified in robots.txt. Providing this, you can prevent the crawlers from indexing your sensitive files and directories. The last and the most advanced suggestion is installing and managing Google honeypots [8] in your system and trying to figure out the behaviour of attackers before they deal with your *real* system.

References

[1] Google Hacking Database.
http://johnny.ihackstuff.com/index.php?module=prodreviews.

[2] GooLink- Google Hacking Scanner.
http://www.ghacks.net/2005/11/23/goolink-scanner-beta-preview/.

[3] SiteDigger v2.0 - Information Gathering Tool.
http://www.foundstone.com.

[4] Johnny Long. Gooscan: Google Security Scanner.
http://johnny.ihackstuff.com/modules.php?op=modload&name=Downloads
&file=index&req=getit&lid=33.

[5] The GNU Privacy Guard. http://www.gnupg.org/(en)/index.html

[6] AxCrypt File Encryption Software for Windows. http://axcrypt.axantum.com

[7] Robots Exclusion Standard. http://en.wikipedia.org/wiki/Robots.txt

[8] Google Hack Honeypot Project. http://ghh.sourceforge.net

[9] Kerberos:The Network Authentication Protocol. http://web.mit.edu/kerberos/

Trusted Computing mit Open Source Software

Heiko Stamer

Universität Kassel, Fachbereich Mathematik/Informatik
Heinrich-Plett-Straße 40, D-34132 Kassel
stamer@theory.informatik.uni-kassel.de

Das Thema *Trusted Computing* wird meist sehr kontrovers diskutiert. Häufig sind Fakten, Vermutungen und persönliche Meinungen stark miteinander vermengt, so dass sich kaum eine neutrale Sichtweise ausmachen läßt. Trotz aller Bedenken, beispielsweise hinsichtlich einschränkender Techniken wie Digital Rights Management (DRM) o. ä., gibt es mittlerweile im Open Source Bereich einige interessante Projekte, welche oft von Industrie oder Wissenschaft (FP6) gefördert werden:

- **TPMDD [Hal06]**
  Gerätetreiber für TPM-Hardware im Linux-Kern (IBM)
- **TrouSerS [Yod06]**
  Softwareschnittstelle für Anwendungsprogramme (IBM)
- **jTSS Wrapper [Win06]**
  Java-Schnittstelle für Anwendungsprogramme (TU Graz)
- **Trusted GRUB [Stü06]**
  Sicheres Booten (Ruhr-Uni Bochum)
- **TPM Emulator [Str06]**
  Softwarebasierter Emulator für TPM-Hardware (ETH Zürich)

Der Vortrag soll einen groben Überblick zur Trusted Computing Initiative der TCG geben und etwas genauer die Zielsetzung, den momentanen Status sowie weiterführende Anwendungsmöglichkeiten der oben genannten Open Source Projekte beschreiben. Zuerst werden wir die kryptographischen Komponenten der TPM-Hardware kennenlernen und einige Anwendungsszenarien (z. B. Remote Attestation) skizzieren. Danach schauen wir uns das Zusammenspiel von Hardware (TPM [TCGa]), Software (TSS [TCGb], Applikation) und Benutzer (Owner/User) anhand einiger konkreter Beispiele an. Weiterhin interessiert uns dabei, welche fortgeschrittenen kryptographischen Anwendungen sich mithilfe von Trusted Computing realisieren lassen.

Zum Abschluß ist eine kurze Demonstration aktueller Software (TPM-Tools, TrouSerS, TPM-Emulator) und eine Diskussion geplant.

Literatur

[TCGa] Trusted Computing Group. TPM Specification. Version 1.2rev94, 2006.

[TCGb] Trusted Computing Group. TSS Specification. Version 1.2, 2006.

[DW06] Wilhelm Dolle, Christoph Wegener. Trusted Computing für Linux: Stand der Dinge. Linux-Magazin 04/2006.

[Str06] Mario Strasser, et al. TPM-Emulator Project. Release 0.4, 2006.
http://tpm-emulator.berlios.de/

[Hal06] Kylene Hall, et al. Linux TPM Device Driver. Kernel 2.6.17, 2006.
http://tpmdd.sourceforge.net/

[Yod06] Kent Yoder, et al. TrouSerS Project. Release 0.2.6, 2006.
http://trousers.sourceforge.net/

[Win06] Thomas Winkler, et al. IAIK/OpenTC jTSS Wrapper. Release 0.2, 2006.
http://trustedjava.sourceforge.net/

[Stü06] Christian Stüble, et al. TrustedGRUB. Release 0.8.1, 2004.
http://www.prosec.rub.de/trusted_grub_details.html

http://KryptoTag.de

Der Kryptotag ist eine zentrale Aktivität der GI-Fachgruppe „Angewandte Kryptologie“. Er ist eine wissenschaftliche Veranstaltung im Bereich der Kryptologie und von der organisatorischen Arbeit der Fachgruppe getrennt. Grundgedanke des Kryptotages ist, dass er inklusive Anreise wirklich nur einen Tag dauert und Nachwuchswissenschaftlern, etablierten Forschern und Praktikern auf dem Gebiet der Kryptologie die Möglichkeit bieten, Kontakte über die eigene Universität hinaus zu knüpfen.

Die Vorträge können ein breites Spektrum abdecken, von noch laufenden Projekten, die ggf. erstmals einem breiteren Publikum vorgestellt werden werden, bis zu abgeschlossenen Forschungsarbeiten, die zeitnah auch auf Konferenzen präsentiert wurden bzw. werden sollen oder einen Schwerpunkt der eigenen Diplomarbeit oder Dissertation bilden. Die eingereichten Abstracts werden gesammelt und als technischer Bericht veröffentlicht. Es handelt sich damit um eine zitierfähige Arbeit. Sie können von den Seiten der Fachgruppe herunter geladen werden.

Geplante Kryptotage

5. Kryptotag am 11. September 2006 (Einreichung: 11. August 2006, Anmeldung: 7. September 2006). Universität Kassel, Arbeitsgruppe Theoretische Informatik. Kontakt: Heiko Stamer.
6. Kryptotag im Februar 2007. Universität des Saarlandes, Information Security and Cryptography Group und Sirrix AG. Kontakt: Michael Backes und Ammar Alkassar.

Bisherige Kryptotage

1. Kryptowochenende am 1.–2. Juli 2006 Tagungszentrum Kloster Bronnbach der Universität Mannheim. Kontakt: Frederik Armknecht und Dirk Stegemann. 14 Einreichungen und 21 angemeldete Teilnehmer.
4. Kryptotag am 11. Mai 2006. Ruhr Universität Bochum, Horst-Görtz Institut. Kontakt: Ulrich Greveler. 10 Einreichungen und 32 angemeldete Teilnehmer.
3. Kryptotag am 15. September 2005. Technische Universität Darmstadt, Theoretische Informatik. Kontakt: Ralf-Philipp Weinmann. 13 Einreichungen und 35 angemeldeten Teilnehmer.
2. Kryptotag am 31. März 2005. Universität Ulm, Abteilung für Theoretische Informatik. Kontakt: Wolfgang Lindner und Christopher Wolf. 10 Einreichungen und 26 angemeldeten Teilnehmer.
1. Kryptotag am 1. Dezember 2004. Universität Mannheim, Lehrstuhl für Theoretische Informatik. Kontakt: Stefan Lucks und Christopher Wolf. 15 Einreichungen und 37 angemeldeten Teilnehmer.

Innerhalb der Fachgruppe für Angewandte Kryptologie sind Stefan Lucks (Universität Mannheim) und Christopher Wolf (École Normale Supérieure, Paris) verantwortlich für die Organisation der Kryptotage. Für evtl. Rückfragen bitte an sie wenden.