Signaturtransformationen

Ruhr-Universität Bochum

Christian Mainka

25. Januar 2011

Inhaltsverzeichnis

1 Einleitung

Digitale Signaturen gehören zu den wichtigsten Primitiven der modernen Kryptographie. Allerdings stellt die Standard-Definition für Sicherheit sehr hohe Anforderungen an ein Signatursystem. Dies macht es äußerst schwierig auf direktem Weg neue Verfahren zu entwickeln, die diesen Anforderungen genügen. Deutlich simpler ist es hingegen Signaturverfahren zu entwickeln, die weniger hohe Sicherheitsanforderungen haben.

Die Idee von Signaturtransformationen ist es, ein Signaturverfahren zu nehmen, welches sehr geringe Sicherheitsanforderungen erfüllt und dieses mit Hilfe eines generischen Algorithmus möglichst effizient in ein neues Verfahren umzuwandeln, welches gegen stärkere Angreifer sicher ist. Dabei wird das neue Verfahren zwar im allgemeinen komplexer werden also das originale Verfahren, was Speicher und Berechenbarkeit betrifft, allerdings kann so die Sicherheit bewiesen und gewährleistet werden.

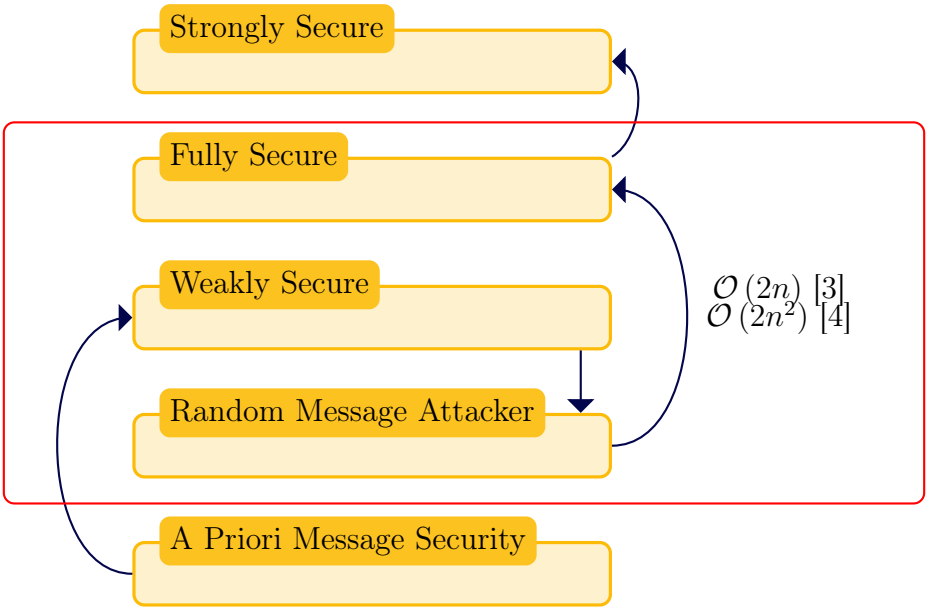


Abbildung 1: Das Ziel der Signaturtransformationen ist es, ein Verfahren, welches resistent gegen einen schwachen Angreifer ist, so zu transformieren, dass ein neues Verfahren, welches sicher gegen einen stärkeren Angreifer ist, entsteht.

Abbildung 1 veranschaulicht das Ziel der Signaturtransformationen: Es wird ein Weg geschaffen, ein *A Priori Message Security* sicheres System bis hin zu einem *Strongly*

Secure sicheren Signaturverfahren zu transformieren. Dabei stellt jeder Pfeil ein solches Transformationsverfahren dar. Die Transformation von *A Priori Message Security* hin zu einem *Weakly Secure* System wurde von Brakerski und Kalai vorgestellt [2], ein Verfahren zur Transformation von einem *Fully Secure* System zu einem *Strogly Secure* System von Huang et. al. [1].

Diese Ausarbeitung beschäftigt sich mit der mittleren Teil der Abbildung und stellt ein Verfahren vor, um ein System, welches sicher für *Random Message Attacker* ist, in ein *Fully Secure* System zu überführen [1] . Ein solches Verfahren wurde bereits von Even, Goldreich und Micali [4] gezeigt. Jedoch geht die Transformation quadratisch mit dem Sicherheitsparameter sowohl in die Rechen- als auch in die Speicherkomplexität ein und ist daher für die Praxis ungeeignet. Die hier vorgestellte Transformation beruht auf dem Ansatz von Cramer und Pedersen [3] und steigert die Komplexität nur um den Faktor zwei, was insbesondere bedeutet, dass nur eine lineare Komplexitätszunahme benötigt wird und das Verfahren somit auch in der Realität benutzt werden kann.

[1] Die Rücktransformation von *Weakly Secure* zum *Random Message Attacker* wird nicht in dieser Ausarbeitung nicht gezeigt. Jedoch ist leicht zu erkennen, dass ein *Random Message Attacker* als Spezialfall eines *Weakly Secure* Systems angesehen werden kann.

2 Grundlagen

Dieses Kapitel führt die benötigten Grundlagen ein. Zunächst wird die

Definition einer Signatur eingeführt und zusätzlich eine weitere wichtige Grundlage für Sicherheitsbeweise in der Kryptographie: die vernachlässigbare Funktion. Im zweiten Teil werden Sicherheitsmodelle für Signaturen vorgestellt. Im Allgemeinen unterscheidet man dabei zwei Dinge: Die Fähigkeiten eines Angreifers und das Ziel, welches dieser erreichen soll.

2.1 Definitionen

2.1.1 Signatur

Definition 1

Ein Signaturverfahren Σ ist ein Tupel $(k, \mathcal{M}, \mathcal{G}, \sigma, \mathcal{V})$ mit:

Sicherheitsparameter k

Nachrichtenraum $\mathcal{M} = \{0,1\}^n$

Initialisierung $\mathcal{G}: \ (pk, sk) \leftarrow 1^k$

Signieren $\sigma \leftarrow \sigma_{sk}(m)$ *für* $m \in \mathcal{M}$

Verifizieren $b = \mathcal{V}_{pk}(m, \sigma) = \begin{cases} 1 & \sigma \text{ gültig} \\ 0 & \text{sonst} \end{cases}$

Ein Signaturverfahren besteht aus einem Sicherheitsparameter k und einem Nachrichtenraum $\mathcal{M}$. Zusätzlich werden drei Algorithmen benötigt:

1. Der Initialisierungsalgorithmus $\mathcal{G}$ (Generator) erstellt ein neues Schlüsselpaar, bestehend aus einem privaten Schlüssel sk und einem öffentlichen Schlüssel pk.
2. Der Signieralgorithmus $\sigma_{sk}(m)$ erstellt eine Signatur σ unter Verwendung des privaten Schlüssels sk für eine Nachricht $m \in \mathcal{M}$. Der Algorithmus ist im allgemeinen nicht deterministisch, d.h. $\sigma_{sk}(m) =: \sigma \neq \tilde{\sigma} := \sigma_{sk}(m)$.

3. Der Verfizieralgorithmus ist deterministisch und gibt ein Bit $b = \mathcal{V}_{pk}(m, \sigma)$ aus. Wenn $b = 1$ gilt, ist σ eine gültige Signatur für die Nachricht m, sonst ist $b = 0$.

In der Literatur findet man auch häufig die Definition einer Signatur als dreier Tupel $(\mathcal{G}, \sigma, \mathcal{V})$. Dabei wird der Sicherheitsparameter k sowie der Nachrichtenraum $\mathcal{M}$ implizit gesetzt. Da aber Beide für die in Kapitel 3 vorgestellte Transformation von Bedeutung sind, werden sie hier mit aufgeführt.

2.1.2 Vernachlässigbare Funktion

Definition 2

Eine Funktion $f : \mathbb{N} \to \mathbb{R}$ heißt vernachlässigbar in k ($\text{negl}(k)$), falls für jedes Polynom p vom Grad maximal k ein $N \in \mathbb{N}$ existiert, so dass für alle $n \geq N$ gilt: $f(n) < \frac{1}{p(n)}$

Eine vernachlässigbare Funktion strebt also schneller gegen Null als das Inverse eines jeden Polynoms. In der Kryptographie findet diese Definition eine große Bedeutung in der asymptotischen Betrachtung für *Computational Security* [5, Seite 50] um sehr geringe Wahrscheinlichkeiten zu betrachten.

Beispiel 1

Um ein besseres Gefühl für obige Definition zu erhalten, seien hier einige Beispiele für vernachlässigbare und nicht vernachlässigbare Funktionen genannt:

- ▷ *Vernachlässigbare Funktionen:* $\frac{1}{2^n}$, $\frac{1}{2^{\sqrt{n}}}$, $\frac{1}{n^{\frac{\log n}{\log \log n}}}$
- ▷ *Nicht vernachlässigbare Funktionen:* $\frac{1}{n^2}$, $\frac{1}{\log n}$

Für den Sicherheitsbeweis in Kapitel 3.3.1 werden außerdem einige Rechenregeln benötigt: Seien f_1, f_2 vernachlässigbare Funktionen in k. Dann gilt:

1. Die Addition von zwei vernachlässigbaren Funktionen (in k) ist wieder vernachlässigbar (in k), d.h. es gilt: $f_1 + f_2 = \text{negl}(k)$
2. Sei $q(n)$ ein beliebiges (nicht vernachlässigbares) Polynom vom Grad k, dann ist die Multiplikation von $q(n)$ mit einer vernachlässigbaren Funktion in k erneut vernachlässigbar, d.h. es gilt: $q(n) \cdot f_1 = \text{negl}(k)$

2.2 Sicherheitseigenschaften von Signaturen

Sehr lange hat man Sicherheit von kryptographischen Primitiven nur dadurch erreicht, dass man ein System sicher gegen einen bestimmten Angreifer gemacht hat. Es gab ein System, z.B. ein Signaturverfahren, welches sicher gegen einen Angreifer $\mathcal{A}_1$ war. Wenn nun ein neuer Angreifer $\mathcal{A}_2$ entdeckt wurde, musste das System umgebaut werden, so dass es zusätzlich auch sicher gegen $\mathcal{A}_2$ ist. Dies bringt das Problem mit sich, dass man ein Verfahren nur sicher gegen bekannte Angreifer machen konnte.

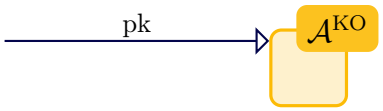
In der modernen Kryptographie nutzt man *Reduktionsbeweise* um zu zeigen, dass ein System gegen jeden nur möglichen Angreifer sicher ist. Um dies zu verwirklichen, nimmt man an, es existiert ein Angreifer $\mathcal{A}$, der ein System brechen kann. Nun benutzt man $\mathcal{A}$, um eine Voraussetzung gegen das System zu brechen, was zu einem Widerspruch führt. Dies zeigt, dass ein solcher Angreifer – unabhängig davon, was dieser tut oder tun kann – überhaupt nicht existieren kann, da falls ein solcher Angreifer existieren würde, eine gegebene Voraussetzung gebrochen werden könnte.

Dazu unterscheidet man zweierlei:

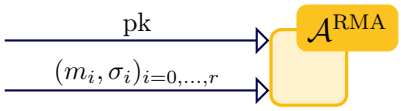
1. Einen Angreifer, der bestimmte Fähigkeiten hat – d.h. er kennt z.B. Klartext-Chiffretextpaare, bzw. Nachrichten- Signaturpaare oder hat Zugriff auf ein Orakel (Kapitel 2.2.1).
2. Der Angreifer muss ein gewisses Sicherheitsziel brechen, z.B. *irgendeine* gültige Signatur für eine *beliebige* Nachricht fälschen (Kapitel 2.2.2).

2.2.1 Angreifertypen

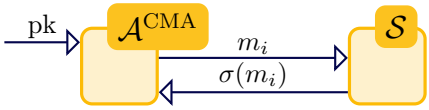
Ein Angreifer auf ein Signaturverfahren Σ kann verschiedene Möglichkeiten und Fähigkeiten haben. Der am schwächsten zu vorstellende Angreifer kennt nur den öffentlichen Schlüssel von Σ (Abbildung 2(a)). Dies ist eine sehr schwache Annahme für einen Angreifer. Ein deutlich stärkerer Angreifer kennt zusätzlich einige zufällige Nachrichten-Signaturpaare (*Random Message Attacker* (RMA), Abbildung 2(b)). Jedoch ist zu beachten, dass dieser Angreifer keine Kontrolle über die erhaltenen Paare hat (z.B. hat er sie durch Mitschneiden des Netzwerkverkehrs abgefangen). Somit ist die logische Konsequenz ein Angreifer, der Zugriff auf ein Signierorakel hat (*Choosen Message Attacker* (CMA), Abbildung 2(c)). Hierbei unterscheidet man in der Literatur oft zwischen einem



(a) Der Key Only Angreifer



(b) Der Random Message Angreifer



(c) Der Choosen Message Angreifer

einfachen Choosen Message Angreifer, der im voraus alle m_i wählt und einem adaptiven Choosen Message Angreifer, der seine Anfragen an das Orakel nacheinander stellt und so dass Wissen aus vorherigen Anfragen nutzen kann.

Der adaptive Choosen Message Angreifer wird im folgenden nur noch als CMA bezeichnet und entspricht den Fähigkeiten, die ein Angreifer eins *Fully Secure* Systems hat.

2.2.2 Sicherheitsziele

[3]

Jeder Angreifer muss ein bestimmtes Ziel erfüllen. Das einfachste und schwächste Ziel ist eine *existentielle Fälschung* (EF). Dies bedeutet, dass der Angreifer für irgendeine *beliebige* Nachricht eine Signatur ausgeben muss. Anders als bei der *selektiven Fälschung* muss der Angreifer diese Nachricht vorher nicht bestimmen, sondern kann diese nach seiner Interaktion (z.B. mit dem Orakel) frei wählen. Dies bedeutet jedoch nicht, dass er Kontrolle über die Nachricht hat – es könnte z.B. die Verknüpfung von vorherigen Werten sein, was bedeutet, das der Angreifer nicht wissen kann, für welche Nachricht er eine Fälschung erstellt. Hat ein Angreifer Kontrolle über die Nachricht, kann diese also wirklich selbst bestimmen, so spricht man von einer *universellen Fälschung*.

Für die Sicherheitsbeweise im folgenden wird nur die existentielle Fälschung betrachtet. Diese ist aus Sicht des Angreifers das am einfachsten zu erreichende Ziel, da ihm keine

Einschränkung für die Nachricht gegeben wird und ist somit auch das erstrebenswerteste Ziel, was ein Signaturverfahren erreichen sollte – wenn ein Angreifer nicht einmal *irgendeine* Nachricht fälschen kann, dann kann er erst recht keine gezielten Nachrichten fälschen.

3 Signaturtransformation

In diesem Kapitel wird die Transformation nach Cramer und Pedersen [3] von einem RMA sicheren Verfahren in ein Fully Secure System vorgestellt. Dazu wird zunächst das Grundkonzept der Transformation in Kaptel 3.1 vorgestellt und der Algorithmus in Kapitel 3.2 vertieft. Im anschließenden Kapitel 3.3 wird zum Abschluss der Sicherheitsbeweis durchgeführt.

3.1 Grundkonzept

Die Grundidee der hier vorgestellten Signaturtransformation zeigt Abbildung 2:

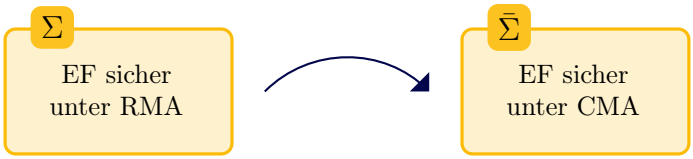


Abbildung 2: Grundidee der Signaturtransformation

Sei $\Sigma = (k, \mathcal{M}, \mathcal{G}, \sigma, \mathcal{V})$ ein nicht existentiell fälschbares Verfahren unter RMA. Konstruiere daraus ein neues Verfahren $\bar{\Sigma} = (\Sigma_1, \Sigma_2)$, welches zwei Instanzen von Σ benutzt und nicht existentiell fälschbar unter CMA ist. Insgesamt ist dass Verfahren $\bar{\Sigma}$ dann gegen den stärkster Angreifer unter der schwächste Annahme sicher.

Um diese Idee zu vertiefen, betrachten wir erneut Abbildung 1 und fügen die Angreifer und deren Ziele wie in Abbildung 3 gezeigt hinzu.

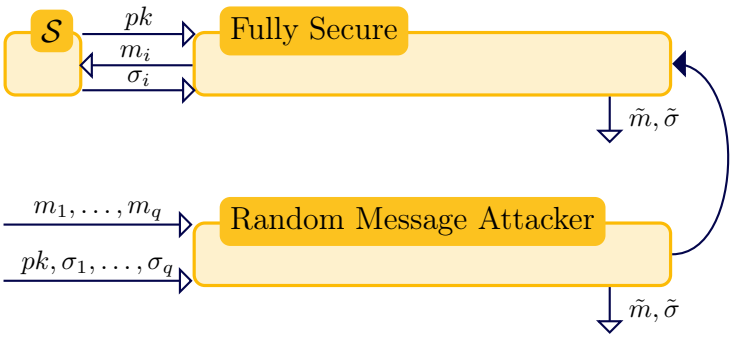


Abbildung 3: Transformation von Σ nach $\bar{\Sigma}$

Als Ergebnis ist zu erkennen, dass beide Angreifer versuchen das selbe Ziel zu brechen (die existentielle Fälschung), dem Angreifer eines *Fully Secure* System allerdings mehr Möglichkeiten zur Verfügung stehen, da er adaptiven Zugriff auf ein Signierorakel hat.

3.2 Die Cramer & Pedersen Transformation

Definition 3

Sei $\Sigma = (k, \mathcal{M}, \mathcal{G}, \sigma, \mathcal{V})$ *ein RMA sicheres Signaturverfahren mit* $\mathcal{M}(k) = \{0,1\}^{t(k)}$ *und Polynom* $t(k)$*, dann konstruiere* $\bar{\Sigma} = (k, \bar{\mathcal{M}}, \bar{\mathcal{G}}, \bar{\sigma}, \bar{\mathcal{V}})$ *wie folgt:*

Nachrichtenraum: *$\bar{\mathcal{M}}(k)$ sei eine Untermenge von $\mathcal{M}(k)$, so dass* $\frac{|\bar{\mathcal{M}}(k)|}{|\mathcal{M}(k)|} = \rho(k) = \mathrm{negl}(k)$

Initialisierung: *Lasse $\mathcal{G}$ zweimal laufen und generiere $(pk_1, sk_1), (pk_2, sk_2)$.*

Signieren: *Sei $m \in \bar{\mathcal{M}}(k)$.*

1. *Wähle $m_1, m_2 \in_R \mathcal{M}(k)$ mit $m_1 \oplus m_2 = m \in \bar{\mathcal{M}}(k)$*
2. *Berechne $\bar{\sigma}(m) = (m_1, m_2, \sigma_1(m_1), \sigma_2(m_2))$*

Verifizieren: $\left(m_1 \oplus m_2 \in \bar{\mathcal{M}}\right) \wedge (\mathcal{V}(m_1, \sigma_1) = 1) \wedge (\mathcal{V}(m_2, \sigma_2) = 1)$

Die Idee der Transformation ist es, Σ zwei mal zu benutzen und so zu verknüpfen, dass ein neues Verfahren $\bar{\Sigma}$ entsteht. Abbildung 4 zeigt eine graphische Veranschauung des Verfahrens.

Der Nachrichtenraum $\bar{\mathcal{M}}$ von $\bar{\Sigma}$ ist eine Untermenge von $\mathcal{M}$ mit vernachlässigbarer Größe (vgl. Abbildung 4 oben), z.B. alle Strings der Länge t mit letzten $t/2$ Bits gleich Null. Um nun eine Nachricht $m \in \bar{\mathcal{M}}$ zu signieren, wird diese in zwei Teile $m_1, m_2 \in \mathcal{M}$ zerlegt, so dass $m_1 \oplus m_2 = m$ gilt. Anschließend werden diese einzeln mit Hilfe der beiden Instanzen von Σ signiert und schließlich zu einem Tupel zusammengefasst, welches die Signatur von $\bar{\Sigma}$ bildet (vgl. Abbildung 4 unten).

Die Komplexität von $\bar{\Sigma}$ ist offensichtlich die zweifache Komplexität von Σ, da für eine Signieroperation in $\bar{\Sigma}$ zwei Signieroperationen in Σ nötig sind (analog bei der Verifikation).

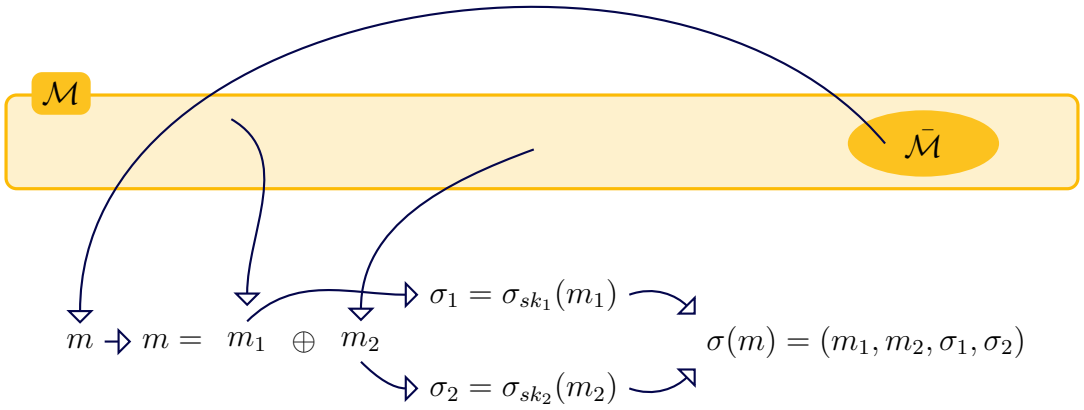


Abbildung 4: Graphische Betrachtung des transformierten Signaturverfahrens $\bar{\Sigma}$

3.3 Sicherheitsbeweis

Im folgenden wird nun der Sicherheitsbeweis für die in Definition 3 aus Kapitel 3.2 vorgestellte Transformation gezeigt. Konkret heißt dies, dass wie in 3.1 eingeführt, ein Verfahren Σ welches nicht existentiell fälschbar unter RMA ist, mithilfe der Konstruktion von Cramer & Pedersen in ein Verfahren $\bar{\Sigma}$, welches nicht existentiell fälschbares unter CMA ist, transformiert wird.

Der Beweis wird dazu in drei Bereiche unterteilt.

1. Aus alten, bekannten Nachrichten/Signaturpaaren von $\bar{\Sigma}$ kann kein gültiges neues Nachrichten/Signaturpaar für $\bar{\Sigma}$ zusammengebaut werden (Kapitel 3.3.1).
2. Es ist nicht möglich, einzelne Signaturteile σ_i des neuen Verfahrens $\bar{\Sigma}$ zu fälschen (Kapitel 3.3.2).
3. Aus 1 und 2 folgt die CMA Sicherheit des Verfahrens $\bar{\Sigma}$ (Kapitel 3.3.3.

3.3.1 Teil 1 von 3

Der erste Teil des Sicherheitsbeweises wird zeigen, dass es nicht möglich ist, eine neue Signatur aus alten bekannten Signaturen zusammen zu setzen. Um ein besseres Verständnis dafür zu bekommen betrachten wir ein einfaches Beispiel:

Beispiel 2

- ▷ *Gegeben:* $m^1 = m_1^1 \oplus m_2^1$ *und* $m^2 = m_1^2 \oplus m_2^2$ *mit* $m^1, m^2 \in \bar{\mathcal{M}}$ *und* $m_1^1, m_2^1, m_1^2, m_2^2 \in \mathcal{M}$.
- ▷ *Seien* $(m_1^1, m_2^1, \sigma_1(m_1^1), \sigma_2(m_2^1))$ *und* $(m_1^2, m_2^2, \sigma_1(m_1^2), \sigma_2(m_2^2))$ *bekannt und gültige Signaturen für* $\bar{\sigma}$
- ▷ *Wenn* $m_1^1 \oplus m_2^2 = \tilde{m} \in \bar{\mathcal{M}}$ *ist, dann ist* $(m_1^1, m_2^2, \sigma_1(m_1^1), \sigma_2(m_2^2))$ *eine gültige Fälschung für eine neue Signatur der Nachricht* $\tilde{m}$ *in* $\bar{\sigma}$

Es ist zu erkennen, dass diese Art der Fälschung allein darauf beruht, ein $\tilde{m} \in \bar{\mathcal{M}}$ unter Verwendung bekannter m^i zu finden. Jedes bekannte Nachrichten/Signaturpaar (m^i, σ^i) mit $m^i \in \bar{\mathcal{M}}$ besteht aus zwei Nachrichtenteilen (m_1^i, m_2^i), wobei $m_{1,2}^i \in \mathcal{M}$ sind (siehe Kapitel 3.2). Sollte nun ein Angreifer r Signaturen kennen und eine Kombination (u, v) für $1 \leq u, v \leq r$ existieren, für die gilt, dass $m_1^u \oplus m_2^v \in \bar{\mathcal{M}}$ ist, können die zugehörigen Signaturen verwendet werden, um eine gültige neue Fälschung zu bauen, wenn $u \neq v$ ist, d.h. wenn die beiden (m_1^u, m_2^v) nicht bereits zum selben $m^u = m^v$ gehören.

Es ergibt sich folgendes Lemma:

Lemma 1

Seien $m^1, \ldots, m^{r(k)} \in \bar{\mathcal{M}}$ *adaptiv gewählt. Seien weiterhin* $(m_1^i, m_2^i) \in_R \mathcal{M}^2$ *zufällig gewählt, sodass stets* $m^i = m_1^i \oplus m_2^i$ *erfüllt ist, dann gilt* $Pr\left[m_1^u \oplus m_2^v \in \bar{\mathcal{M}}\right] = \text{negl}(k)$ *für* $u \neq v$.

Um die Gültigkeit von Lemma 1 zu beweisen wird das Spiel aus Abbildung 5 betrachtet.

Das Spiel zeigt einen Angreifer $\mathcal{A}$, der Zugriff auf ein Signierorakel $\mathcal{S}$ hat (zu erkennen am Index $\mathcal{S}$ im Bild). $\mathcal{A}$ erhält den Sicherheitsparameter k, sowie die Nachrichtenräume $\mathcal{M}$ und $\bar{\mathcal{M}}$ und leitet diese an das Orakel weiter. $\mathcal{S}$ generiert sich ein Schlüsselpaar und sendet den öffentlichen Teil zurück an $\mathcal{A}$. Damit sind die Grundvoraussetzungen für ein Angreifer-Spiel gesetzt – beide Spieler spielen das selbe Spiel.

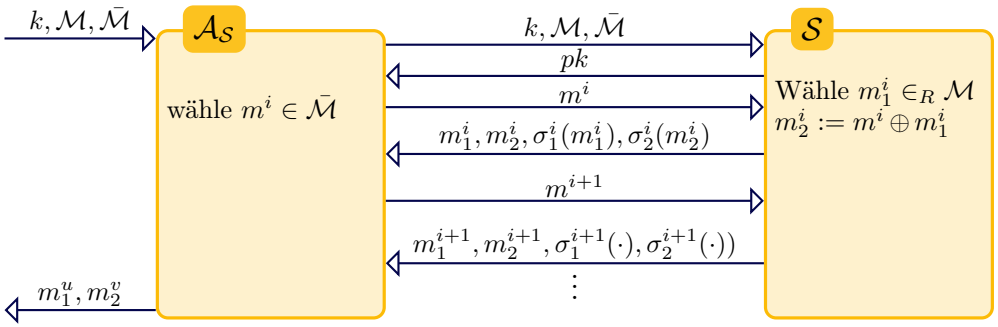


Abbildung 5: Spiel zu Lemma 1

Nun hat $\mathcal{A}$ die Möglichkeit, ein $m^i \in \bar{\mathcal{M}}$ zu wählen und dieses an $\mathcal{S}$ zu senden. $\mathcal{S}$ muss die Anfrage von $\mathcal{A}$ beantworten und wählt dazu zufällig ein $m_1^i \in_R \mathcal{M}$. Den zweiten Teil kann $\mathcal{S}$ leicht errechnen und setzt $m_2^i = m^i \oplus m_1^i$. Somit gilt $(m_1^i, m_2^i) \in \mathcal{M}^2$ und die Bedingung $m^i = m_1^i \oplus m_2^i$ ist ebenfalls erfüllt.

$\mathcal{S}$ sendet nun m_1^i, m_2^i zusammen mit den entsprechend Signaturen (welche für diese Lemma nicht relevant sind, da statistische Fälschungen wie zuvor beschrieben das Ziel sind) an $\mathcal{A}$ zurück.

$\mathcal{A}$ darf nun weitere $m^{i+1}, \ldots, m^{r(k)}$ Anfragen stellen und $\mathcal{S}$ kann diese wie oben gezeigt beantworten. Am Ende kennt $\mathcal{A}$ die Paare $(m_1^1, m_2^1), \ldots, (m_1^{r(k)}, m_2^{r(k)})$ (und die dazugehörigen Signaturen). $\mathcal{A}$ gewinnt das Spiel, wenn er ein paar (u, v) findet, mit $1 \leq u, v \leq r(k)$ und $u \neq v$, so dass die Bedingung $m_1^u \oplus m_2^v \in \bar{\mathcal{M}}$ erfüllt ist.

Die Gewinnchancen von $\mathcal{A}$ hängen einzig und allein von der Zufallswahl der m_i von $\mathcal{S}$ ab. Die Betrachtung von Abbildung 6 verdeutlicht dies:

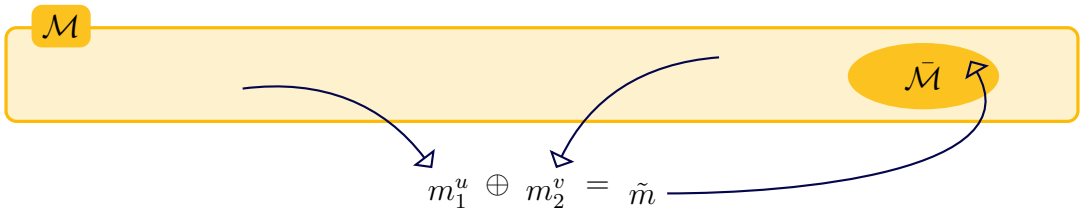


Abbildung 6: Auswertung des Spiels

▷ $\mathcal{A}$ stehen $r(k)$ viele Paare (m_1^i, m_2^i) zur Verfügung.

- ▷ Damit hat $\mathcal{A}$ $r(k) \cdot (r(k) - 1)$ viele Kombinationsmöglichkeiten (m_1^u, m_2^v).
- ▷ Jedes $m_{1,2}^i$ ist zufällig aus $\mathcal{M}$ gezogen.
- ▷ Damit ist jedes $\tilde{m} = m_1^u \oplus m_2^v$ ebenfalls zufällig aus $\mathcal{M}$ gezogen.
- ▷ Die Wahrscheinlichkeit, dass $\tilde{m}$ ein Element aus $\bar{\mathcal{M}}$ ist hängt damit von der Größe, also der Anzahl der Elemente, von $\bar{\mathcal{M}}$ im Verhältnis zu $\mathcal{M}$ ab.
- ▷ Dieses Verhältnis ist nach Definition 3 durch $\rho(k)$ bestimmt.
- ▷ Insgesamt ist die Chance, dass $\tilde{m} \in \bar{\mathcal{M}}$ ist also $r(k)\,(r(k) - 1) \cdot \rho(k)$.
- ▷ $\bar{\mathcal{M}}$ wurde allerdings so gewählt, dass $\rho(k) = \text{negl}(k)$ gilt.
- ▷ Nach den Rechenregeln für vernachlässigbare Funktionen gilt demnach:

$$r(k)\,(r(k) - 1) \cdot \rho(k) = r(k)\,(r(k) - 1) \cdot \text{negl}(k) = \text{negl}(k)$$

Also kann $\mathcal{A}$ das Spiel nur mit vernachlässigbarer Wahrscheinlichkeit gewinnen und es gilt Lemma 1.

3.3.2 Teil 2 von 3

Der zweite Teil des Sicherheitsbeweises wird zeigen, dass es keinen Angreifer $\mathcal{A}$ auf $\bar{\Sigma}$ geben kann, der eine einzelne σ_i Komponente nach einem CMA Angriff für $i = 1, 2$ fälschen kann. Dazu zunächst das Lemma:

Lemma 2

Wenn Σ nicht existentiell fälschbar unter RMA ist kann ein Angreifer $\mathcal{A}$ nur mit vernachlässigbarer Wahrscheinlichkeit ***eine*** *Signatur $\tilde{\sigma}(\tilde{m})$ in Σ_{pk_i} für $\tilde{m} \neq m_i^j,\ j = 1, \ldots, r(k),\ i = 1, 2$ fälschen.*

Es ist zu beachten, dass Lemma 2 nicht gleichzusetzen ist mit der CMA Sicherheit von $\bar{\Sigma}$, da Lemma 2 lediglich verbietet, eine Teilsignatur σ_1 oder σ_2 zu fälschen.

Um die Korrektheit von Lemma 2 zu zeigen, wird eine Standard Simulationstechnik benutzt:

- ▷ Angenommen es gibt einen Angreifer $\mathcal{A}$, der mit nicht vernachlässigbarer Wahrscheinlichkeit Lemma 2 brechen kann.
- ▷ Dann kann man einen Angreifer $\mathcal{A}'$ konstruieren, der mit nicht vernachlässigbarer Wahrscheinlichkeit die RMA Sicherheit von Σ bricht.
- ▷ Dies ist aber nach Voraussetzung nicht möglich.

Diese Beweistechnik nennt man auch Reduktionsbeweis: Man nimmt an, dass ein Angreifer $\mathcal{A}$ auf ein System existiert (ohne zu wissen, was genau dieser tut oder wie er dies tut) und nutzt $\mathcal{A}$ dann, um einen neuen Angreifer $\mathcal{A}'$ zu bauen, der eine Voraussetzung an das System – in diesem Fall die RMA Sicherheit – bricht, was zu einem Widerspruch führt und bedeutet, dass ein solcher Angreifer $\mathcal{A}$ überhaupt nicht existieren kann.

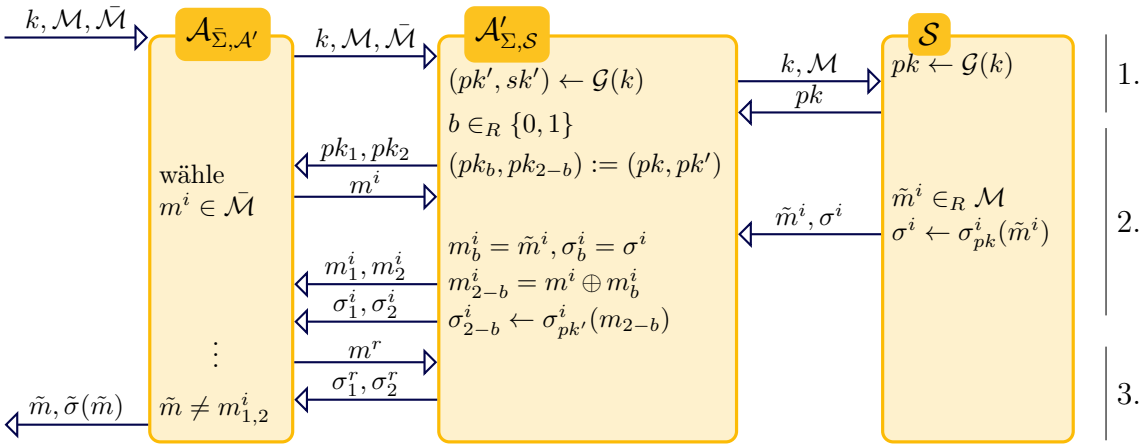


Abbildung 7: Spiel

Um diesen Widerspruch zu verdeutlichen wird das Spiel aus Abbildung 7 betrachtet. Sei $\mathcal{A}$ ein Angreifer der mit nicht vernachlässigbarer Wahrscheinlichkeit $\varepsilon(k)$ Lemma 2 brechen kann und $\mathcal{A}'$ ein Spieler, der Zugriff auf ein Signierorakel $\mathcal{S}$ hat. Betrachte nun die drei Teile des Spiels:

1. Alle Spiele müssen das selbe Spiel spielen und deshalb die nötigen Parameter austauschen. $\mathcal{A}$ ist ein Angreifer auf $\bar{\Sigma}$, erhält die Parameter $k, \mathcal{M}$ und $\bar{\mathcal{M}}$ und leitet diese an $\mathcal{A}'$ weiter. $\mathcal{A}'$ selbst ist der oben erwähnte, konstruierte Angreifer auf Σ und hat seinerseits Zugriff auf ein Signierorakel $\mathcal{S}$, welches ihm zufällige gültige Signaturen in Σ erzeugen kann. $\mathcal{A}'$ sendet also die benötigten Parameter k und $\mathcal{M}$

an $\mathcal{S}$ und erhält den öffentlichen Schlüssel von $\mathcal{S}$. Da $\mathcal{A}'$ selbst fähig seien muss, die CMA Anfragen von $\mathcal{A}$ in $\bar{\Sigma}$ zu beantworten, generiert sich $\mathcal{A}'$ ein Schlüsselpaar (pk', sk'). Nun wählt $\mathcal{A}'$ ein zufälliges $b \in_R \{0, 1\}$ und ordnet dadurch den öffentlichen Schlüssel des Signierorakels pk sowie den eigenen öffentlichen Schlüssel pk' den Schlüsseln (pk_b, pk_{2-b}) zu. Dadurch ist pk_1 mit Wahrscheinlichkeit $\frac{1}{2}$ der öffentliche Schlüssel von $\mathcal{A}'$ oder von $\mathcal{S}$, pk_2 analog. Das Schlüsselpaar (pk_1, pk_2) wird anschließend an $\mathcal{A}$ gesendet.

2. $\mathcal{A}$ darf nun CMA Anfragen stellen und $\mathcal{A}'$ muss diese beantworten. Konkret heißt dies, dass $\mathcal{A}$ ein $m^i \in \bar{\mathcal{M}}$ wählt und dieses als Anfrage an $\mathcal{A}'$ sendet. $\mathcal{A}'$ muss diese Anfrage nun beantworten und lässt sich ein zufälliges Nachricht/Signaturpaar $(\tilde{m}, \sigma^i)$ von $\mathcal{S}$ zusenden. Nun ordnet $\mathcal{A}'$ zunächst $m_b^i = \tilde{m}^i$ und $\sigma_b^i = \sigma^i$ zu, da $pk_b = pk$ dem öffentlichen Schlüssel von $\mathcal{S}$ entspricht. Um die zweite Komponente zu berechnen, wählt $\mathcal{A}'$ zunächst $m_{2-b}^i = m^i \oplus m_b^i$ und kann die passende Signatur anschließend mit Hilfe seines eigenen sk' selbst errechnen. Anschließend sendet $\mathcal{A}'$ die Nachricht $(m_1^i, m_2^i, \sigma_1^i, \sigma_2^i)$ an $\mathcal{A}$ zurück (Aus Platzgründen wurden in der Zeichnung zwei Pfeile verwendet).

3. $\mathcal{A}$ darf nun adaptiv immer neue Nachrichten wählen und $\mathcal{A}'$ kann diese wie in 2. gezeigt beantworten. Am Ende gibt $\mathcal{A}$ eine Fälschung $\tilde{m}, \tilde{\sigma}(\tilde{m})$ aus, wobei $\tilde{m}$ keinem durch CMA Anfragen erhaltenen m_1^i oder m_2^i entspricht.

Auswertung:

- ▷ $\mathcal{A}$ kann Spiel nicht von echtem Signierverfahren $\bar{\Sigma}$ unterscheiden.
- ▷ Angenommen $\mathcal{A}$ kann das Spiel mit nicht vernachlässigbarer Wahrscheinlichkeit $\varepsilon(k)$ gewinnen.
- ▷ Dann könnte der Angreifer $\mathcal{A}'$ genutzt werden, um Signaturen mit Wahrscheinlichkeit $\frac{1}{2}\varepsilon(k)$ von $\mathcal{S}$ zu fälschen, da die Ausgabe von $\mathcal{A}$ mit Wahrscheinlichkeit $\frac{1}{2}$ eine Fälschung für pk_b ist.
- ▷ Diese Wahrscheinlichkeit wäre immer noch nicht vernachlässigbar.
- ▷ Laut Voraussetzung ist Σ aber RMA sicher.
- ▷ Widerspruch zur Voraussetzung.

Es kann kein Angreifer $\mathcal{A}$ existieren, der das Spiel mit nicht vernachlässigbarer Wahrscheinlichkeit gewinnt, also ist die Erfolgswahrscheinlichkeit von $\mathcal{A}$ ebenfalls $\text{negl}(k)$.

3.3.3 Teil 3 von 3

Nun werden Lemma 1 und Lemma 2 genutzt um endgültig zu zeigen, dass $\bar{\Sigma}$ *Fully Secure*, also nicht existentiell fälschbar unter CMA ist.

Lemma 3

Wenn Σ ein nicht existentiell fälschbares Signaturverfahren unter RMA ist, dann ist $\bar{\Sigma}$ nicht existentiell fälschbar unter CMA.

Die Korrektheit des Lemmas folgt nach dem Ausschlussprinzip unter Verwendung der vorherigen Lemmas:

- ▷ Sei $\tilde{m} \in \bar{\mathcal{M}}(k)$ und sei $\tilde{\sigma}(\tilde{m}) = (m_1, m_2, \sigma_1(m_1), \sigma_2(m_2))$ eine Fälschung nach CMA Angriff.
- ▷ Nach Lemma 2 können $\sigma_i(m_i)$ keine Fälschungen sein sondern müssen aus CMA Anfragen stammen.
- ▷ Nach Lemma 1 gilt aber $Pr\left[m_1^u \oplus m_2^v \in \bar{\mathcal{M}}\right] = \text{negl}(k)$ für $u \neq v$.

Da nun weder einzelne Signaturkomponenten gefälscht werden können, noch eine statistische Fälschung durch zusammensetzen von alten bekannten Nachricht/Signaturpaaren möglich ist, muss $\bar{\Sigma}$ CMA sicher sein.

4 Fazit

Der Algorithmus nach Cramer & Pedersen bietet eine effektive Transformation von einem *Weakly Secure* zu einem *Fully Secure* Signaturverfahren. Es funktioniert nach einem simplen Prinzip und nutzt ein *Weakly Secure* Verfahren in geschickter Kombination zweifach und erhöht dadurch die Komplexität ebenfalls nur auf das doppelte, was insbesondere bedeutet, dass das Verfahren in der selben Komplexitätsklasse bleibt und damit auch in der Praxis benutzbar bleibt. Außerdem bietet es die interessante Möglichkeit im Rahmen eines *On/Offline Signaturverfahrens*, wie bereits von Even et. al [4] vorgestellt, einen Time-Memory Trade-Off durchzuführen. Dazu würden beispielsweise eine Reihe von m_1^i, σ_1^i Komponenten offline vorberechnet und gespeichert werden. In der Online Phase würden dann schließlich die m_2^i, σ_2^i Komponenten berechnet werden. Dies Erlaubt eine Signaturberechnung in $\bar{\Sigma}$ ebenso schnell wie in Σ, wobei selbstverständlich die vorberechneten Komponenten einen gewissen Speicherplatz benötigen.

Appendix

Literatur

[1] Bellare, M. und S. Shoup: *Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles*. In: *Proceedings of the 10th international conference on Practice and theory in public-key cryptography*, PKC'07, S. 201–216, Berlin, Heidelberg, 2007. Springer-Verlag, ISBN 978-3-540-71676-1. `http://portal.acm.org/citation.cfm?id=1760564.1760583`.

[2] Brakerski, Z. und Y. T. Kalai: *A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model*. Cryptology ePrint Archive, Report 2010/086, Feb 2010. `http://eprint.iacr.org/2010/086`.

[3] Cramer, R., T. Pedersen, I. x, R. Cramer und T. Pedersen: *Efficient and Provable Security Amplifications*. In: *CS-R9529, Computer Science, Dept. of Algorithms and Architecture, CWI*, S. pages., 1995.

[4] Even, S., O. Goldreich und S. Micali: *On-Line/Off-Line Digital Signatures*, 1994.

[5] Katz, J. und Y. Lindell: *Introduction to modern cryptography*. Chapman & Hall/CRC Cryptography and Network Security. Chapman & Hall/CRC, Boca Raton, FL, 2008, ISBN 978-1-58488-551-1; 1-58488-551-3.