Ruhr-Universität Bochum
Lehrstuhl für Kryptologie und IT-Sicherheit
Prof. Dr. Alexander May
Alexander Meurer, Felix Heuer

RUHR UNIVERSITÄT BOCHUM RUB

Hausübungen zur Vorlesung

Kryptographie 2

SS 2011

Blatt 6 / 22. Juni 2011 / Abgabe 13. Juli, 08:30 Uhr, Kasten NA 02

AUFGABE 1. DCR-Annahme und Faktorisierung. (5 Punkte)
Beweisen Sie: Wenn das DCR-Problem hart ist bzgl $\mathsf{GenModulus}$ (siehe Folie 111), so ist auch Faktorisieren hart bzgl. $\mathsf{GenModulus}$ (siehe Folie 57). Gehen Sie wie folgt vor:

a) Zeigen Sie, dass man bei bekannter Faktorisierung effizient den inversen Isomorphismus $f^{-1} : \mathbb{Z}_{N^2}^* \to \mathbb{Z}_N \times \mathbb{Z}_N^*$ berechnen kann.

b) Zeigen Sie, dass man das DCR-Problem effizient lösen kann, wenn f^{-1} effizient berechenbar ist.

AUFGABE 2. Starke Einwegsignaturen. (5 Punkte)
Informell sprechen wir von einem *starken Einwegsignaturverfahren* $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, falls es keinen Angreifer $\mathcal{A}$ gibt, der eine gültige Signatur σ^* für m^* berechnen kann mit $(m^*, \sigma^*) \neq (m, \sigma)$ wobei σ eine gültige Signatur für eine von $\mathcal{A}$ gewählte Nachricht m ist (d.h. im Gegenteil zum üblichen Einwegsignatur-Spiel (Folie 138) muss nicht zwangsläufig $m^* \neq m$ gelten).

a) Formalisieren Sie die obige informelle Definition, indem Sie ein geeignetes Spiel $\mathsf{StrongForge}_{\mathcal{A},\Pi}^{\text{einweg}}$ angeben und den Vorteil von $\mathcal{A}$ definieren.

b) Nehmen Sie die Existenz von Einwegfunktionen an und konstruieren Sie eine spezielle Einwegfunktion f, für welche das Lamport-Signaturschema *kein* starkes Einwegsignaturverfahren ist. Begründen Sie sowohl die Einwegeigenschaft der von Ihnen gewählten Funktion f und geben Sie einen Fälscher an.

c) Zeigen Sie, dass das Lamport-Verfahren ein starkes Einwegsignaturverfahren ist, wenn man für f eine *Einwegpermutation* verwendet.

Wir haben in der Präsenzübung eine exaktere Analyse der Reduktion zum FDH-RSA Signaturschema diskutiert, in welcher man die Orakelanfragen des Angreifers $\mathcal{A}$ in Signatur- und Random-Oracle Anfragen unterscheidet. Wir haben die Anzahl der jeweiligen Anfragen mit q_{sig} bzw. q_H bezeichnet. Diese Unterteilung ist sinnvoll, denn in der Praxis hat ein Angreifer $\mathcal{A}$ leicht die Möglichkeit, eine große Menge von Anfragen an die Hashfunktion H zu stellen (bspw. $q_H \approx 2^{50}$), da diese öffentlich bekannt ist. Hingegen ist es deutlich unwahrscheinlicher, dass er eine sehr große Menge Signaturen für adaptiv von ihm gewählte Nachrichten erhält (bspw. kalkuliert man oftmals mit $q_{\text{sig}} \approx 2^{30}$).
Bei genauerer Betrachtung des Sicherheitsbeweises von FDH-RSA kann man folgende verfeinerte Variante des Satzes "CMA-Sicherheit von FDH-RSA" (Folie 130) beweisen.

Theorem 1. Wenn das RSA-Problem (t', ϵ')-hart ist, dann ist FDH-RSA (t, ϵ)-sicher mit $t = t' - (q_H + q_{\text{sig}} + 1) \cdot \mathcal{O}(n^3)$ und $\epsilon = (q_H + q_{\text{sig}}) \cdot \epsilon'$ wobei $n = \log_2(N)$ die Bitlänge des RSA-Moduls bezeichnet.

Hierbei nennen wir das RSA-Problem (t', ϵ')-hart, falls kein Algorithmus $\mathcal{A}$ mit Laufzeit $\leq t'(n)$ das RSA-Problem mit Erfolgswahrscheinlichkeit $\geq \epsilon'(n)$ löst (siehe Folie 37).
In der folgenden Aufgabe wollen wir nun einen alternativen Beweis für die CMA-Sicherheit von FDH-RSA finden, der bei gleichem t ein besseres, d.h. kleineres, ϵ erreicht. In der Praxis würde dies bei gleicher Parameterwahl ein höheres Sicherheitslevel garantieren (oder im Umkehrschluss bei kleinerer Parameterwahl, also höherer Effizienz, ein identisches Sicherheitslevel ermöglichen).

Theorem 2. Wenn das RSA-Problem (t', ϵ')-hart ist, dann ist FDH-RSA (t, ϵ)-sicher mit $t = t' - (q_H + q_{\text{sig}} + 1) \cdot \mathcal{O}(n^3)$ und

$$\epsilon = \frac{1}{\left(1 - \frac{1}{q_{\text{sig}}+1}\right)^{q_{\text{sig}}+1}} \cdot q_{\text{sig}} \cdot \epsilon'$$

wobei $n = \log_2(N)$ die Bitlänge des RSA-Moduls bezeichnet. Für große q_{sig} gilt $\epsilon \approx \exp(1) \cdot q_{\text{sig}} \cdot \epsilon'$.

AUFGABE 3. Effizienteres FDH-RSA. (5 Punkte)
Beweisen Sie Theorem 2.

Programmieren Sie hierzu das Random-Oracle H wie folgt: Mit Wahrscheinlichkeit $0 < p < 1$ setzen Sie $H(m_i) := r_i^e \bmod N$ für $r_i \in_R \mathbb{Z}_N^*$. Mit Wahrscheinlichkeit $(1 - p)$ setzen Sie $H(m_i) := y \cdot r_i^e \bmod N$ für $r_i \in_R \mathbb{Z}_N^*$. Hierbei ist $y \in_R \mathbb{Z}_N^*$ die Challenge für das RSA-Problem.
Untersuchen Sie dann, für welche m_i man Signaturen simulieren kann und unter welcher Bedingung man aus einer gültigen Fälschung (m^*, σ^*) eine Antwort $x = y^{1/e} \bmod N$ berechnen kann. Daraus können Sie schliesslich eine von p abhängige Wahrscheinlichkeit $\alpha(p)$ berechnen, mit welcher man das RSA-Problem löst. Geben Sie dann ein optimale Wahl von p an, so dass $\alpha(p)$ maximal wird.

AUFGABE 4. Online / Offline Signaturen. (5 Punkte)
Digitale Signaturen sind oft relativ "teuer" (in Bezug auf die Rechenzeit). Die Idee von Online / Offline Signaturen besteht darin, den Signierprozeß in zwei Komponenten aufzuteilen. Während einer *Offline*-Phase wird eine Teilsignatur σ_1 vorberechnet, ohne die zu signierende Nachricht zu kennen. Anschliessend wird in der *Online*-Phase aus den Informationen σ_1 und m eine zweite Teilsignatur σ_2 berechnet und zur gesamten Signatur σ zusammengefasst.
Sei nun $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ ein sicheres Signaturschema und sei $\Pi' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vrfy}')$ ein sicheres One-Time Signaturschema. Wir konstruieren daraus ein Online / Offline Schema $(\widetilde{\mathsf{Gen}}, \widetilde{\mathsf{Sign}}, \widetilde{\mathsf{Vrfy}})$ wie folgt:

- $\widetilde{\mathsf{Gen}}(1^n)$ gibt ein Schlüsselpaar gemäß Π aus, d.h. $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^n)$.
- $\widetilde{\mathsf{Sign}}_{\mathsf{sk}}(m)$
  - Offline-Phase: Wähle $(\mathsf{sk}', \mathsf{pk}') \leftarrow \mathsf{Gen}'(1^n)$ und berechne $\sigma_1 = \mathsf{Sign}_{\mathsf{sk}}(\mathsf{pk}')$.
  - Online-Phase: Berechne $\sigma_2 = \mathsf{Sign}'_{\mathsf{sk}'}(m)$.

  Gib die Signatur $\sigma = (\sigma_1, \sigma_2, \mathsf{pk}')$ aus.

a) Geben Sie eine sinnvolle Verifikation $\widetilde{\mathsf{Vrfy}}$ an und begründen Sie die Korrektheit.

b) Beweisen Sie die CMA-Sicherheit des Verfahrens.

Hinweis: Hierbei ist es hilfreich, ein Ereignis BadKey einzuführen, welches widerspiegelt, ob (mindestens) eine angefragte Nachricht m_i eine Signatur $\sigma_i = (\sigma_{1,i}, \sigma_{2,i}, \mathsf{pk}'_i)$ erhalten hat, so dass $(\mathsf{pk}')^* = \mathsf{pk}'_i$ gilt. Hierbei ist $\sigma = (\sigma_1^*, \sigma_2^*, (\mathsf{pk}')^*)$ die von $\mathcal{A}$ ausgegebene Fälschung für m^*.