Sicherheit durch Padding

„Padding“ eigentlich besser „Encoding“ oder „Embedding“ ...
Plain RSA ist unsicher, im folgenden betrachten wir Modifikationen oder Padding schemes, welche es sicher machen. Zur Erinnerung:

Sicherheitsstufen:

- OW Sicherheit (one-wayness)
- IND Sicherheit (indistinguishability)
- NM Sicherheit (non-malleability)

Angreifertypen:

- CPA (chosen plaintext attack). Angreifer passiv.
- CCA1 (non-adaptive chosen ciphertext) und CCA2 (adaptive chosen ciphertext attack). In beiden Fällen Angreifer aktiv.

Thm: IND-CCA2 ist äquivalent zu NM-CCA2.

Bemerkung

Es gibt auch den Begriff der semantischen Sicherheit:

Was immer ein passiver Angreifer (über den Klartext) aus dem Chiffretext ausrechnen kann, kann er auch ohne den Chiffretext ausrechnen (in erwartet polynomieller Laufzeit, über alle Wahrscheinlichkeitsverteilungen der Klartexte).

Thm: Semantische Sicherheit ist äquivalent zur Sicherheitseigenschaft IND.

Zwischen beiden Konzepten liegt folgender Angriff (A):

Für einen Chiffretext c zu entscheiden, ob er zu einem gewählten, bekannten Klartext m gehört, oder zu einer (unbekannten) zufälligen Nachricht.

Bemerkung

A $\Rightarrow$ IND: Wir können A mit Wahrscheinlichkeit $1/2+\varepsilon$ lösen. Um c den Nachrichten m_1, m_2 zuzuordnen, entscheiden wir, ob c zu m_1 gehört (ohne m_2 zu beachten). Erfolgswahrscheinlichkeit $1/2+\varepsilon$.

IND $\Rightarrow$ A: Wir können IND mit Wahrscheinlichkeit $1/2+\varepsilon$ lösen. Um zu entscheiden, ob c zu m gehört, wähle zufälliges m'. Dann entscheiden, ob c zu m oder m' gehört. Liefert dies m, dann Ausgabe c gehört zu m. Liefert dies einen Fehler, dann Ausgabe c gehört nicht zu m. Liefert dies m', so Ausgabe c gehört zu m oder nicht mit Wahrscheinlichkeit $1/2$. Erfolgswahrscheinlichkeit größer ca. $(1/2)(1/2+\varepsilon)+1/4 = 1/2+\varepsilon/2$.

A $\Leftrightarrow$ semantische Sicherheit: A bedeutet, daß man Chiffretexte zu bekannten Nachrichten durch Chiffretexte beliebiger zufälliger Nachrichten ersetzen kann. Da jeder Chiffretexte ausrechnen kann, ergibt sich semantische Sicherheit.

Sicherheit von Plain RSA und Rabin

Wr nehmen an, daß das RSA und Rabin Problem schwierig ist.

Plain RSA dann OW-CPA sicher, auch OW-CCA1 sicher?
(Invertieren der RSA Funktion impliziert Faktorisieren?).
Rabin ist OW-CPA sicher, aber nicht OW-CCA1 sicher.
(Wurzelziehen impliziert Faktorisieren).

Sind nicht IND: Gegeben c, m_1, m_2, verschlüssele $c_i = m_1^e \bmod n$ und gib i mit $c_i = c$ zurück.

Sind nicht NM: $c' = cs^e$ ist Chiffretext von ms, wenn $c = m^e \bmod n$.

Sind nicht PA (plain text aware): Die RSA Funktion ist bijektiv, jedes $c \in \mathbb{Z}/n\mathbb{Z}$ ist Chiffretext. Ähnlich für die Rabin Funktion.

CPA Sicherheit zu wenig

CPA Sicherheit ist im allgemeinen viel zu wenig, Beispiel Bleichenbachers Angriff (1998):

Problematische Verwendung von RSA in PKCS #1 v1.5, insbesondere SSL v3.0 ist nicht CCA1 sicher (`https://...`)!

Gültige Klartexte m sind von der Form $(00||02||PS||00||D)$, für einen Padding String PS und Daten D. Es werden nur die ersten beiden Bytes getestet.

Angriff:

- Schicke Chiffretexte $c' = cs^e \bmod n$ zum Entschlüsseln.
- Wenn Fehler, dann sind die ersten Bytes ungleich $00||02$, sonst gleich. Erhalten also, ob sm in einem gewissen Intervall liegt oder nicht.
- Ähnliches Vorgehen wie bei der Intervallfunktion f liefert m.

CPA Sicherheit zu wenig

Im SSL v3.0 wird noch die Versionsnummer den Daten D vorangestellt.
Dies scheint Bleichenbachers Angriff zu erschweren.

Aber: Umgang mit ungültigen Chiffretexten ist nicht spezifiziert bzw. ist schwierig.

- Server überprüf(t)en Versionsnummer nicht.
- Server test(et)en die Bedingungen und liefer(te)n Fehlermeldung, wo das Problem liegt.

IND-CCA2 Sicherheit

Wir nehmen vereinfachend an, daß wir eine Einwegpermutation $f : \{0,1\}^n \to \{0,1\}^n$ mit Falltür betrachten (z.B. die RSA Funktion).

Wir betrachten jetzt Kryptosysteme, die einen Sicherheitsbeweis im Zufallsorakelmodell haben.

- Ergibt keine Sicherheit in der „real world“, wo eine durch ein Zufallsorakel modellierte Hashfunktion durch eine konkrete Hashfunktion (SHA-1) ersetzt wird.
- Aber Sicherheit gegen die Klasse von Angriffen, die keine speziellen Eigenschaften der Hashfunktionen ausnutzen.
- Solche treten in der Praxis besonders häufig auf.

Erste Variante

Sei $H : \{0,1\}^* \to \{0,1\}^n$ ein Zufallsorakel.

Verschlüsseln von $m \in \{0,1\}^n$:

- Wähle $r \in \{0,1\}^n$ zufällig.
- Berechne $y = f(r)$ und $z = m \oplus H(r)$.
- Ausgabe des Chiffretexts (y, z).

Entschlüsseln von (y, z):

- Berechne $r = f^{-1}(y)$ mit der Falltürinformation.
- Berechne $m = z \oplus H(r)$.
- Ausgabe der Nachricht m.

Erste Variante

Thm: Die erste Variante ist IND-CPA sicher.

Bew: Sei A ein Algorithmus für den Angriff A, welcher mit Wahrscheinlichkeit $1/2+\varepsilon$ Erfolg hat. Gegeben sei ein zufällig gewähltes y. Wir wählen m', z zufällig und rufen A mit m' und (y,z) auf. Hier ist (y,z) ein gültiger Chiffretext, nur sind uns das entsprechende $r = f^{-1}(y)$ und $m = z \oplus H(r)$ unbekannt. Die Orakelanfragen an H von A werden von uns beantwortet. Mit Wahrscheinlichkeit $\geq \varepsilon$ stellt A die Orakelanfrage $H(r)$, sonst hätte A keine Information über $H(r)$ und damit m, und hätte folglich Erfolgswahrscheinlichkeit $1/2$. Gegeben y erhalten wir also r aus dem Orakel mit Erfolgswahrscheinlichkeit $\geq \varepsilon$. $\square$

Allg. Beweisidee: Wir implementieren H und benutzen A dazu, ein schwieriges Problem zu lösen, hier Urbilder unter f auszurechnen.

Erste Variante

Thm: Die erste Variante ist IND-CCA1 sicher.

Bew: Der Beweis ist wie eben, nur müssen wir zusätzlich die Entschlüsselungsanfragen von A simulieren. Dies geschieht wie folgt. Ist $H^*(y)$ undefiniert, definieren wir es als zufälligen Wert. Auf Eingabe von (y,z) antworten wir mit $m = z \oplus H^*(y)$. Auf eine Hashanfrage $H(r)$ überprüfen wir, ob $H(r)$ bereits definiert ist. Ist es undefiniert, setzen wir $H(r) = H^*(f(r))$.

Sei y zufällig gegeben. In der Fragephase erhalten wir eine Anfrage zur Entschlüsselung von (y,z) mit insignifikanter Wahrscheinlichkeit. Folglich ist $H(r)$ mit $r = f^{-1}(y)$ in der Aufgabenphase undefiniert, so daß r von A wie oben berechnet werden kann. $\square$

Algemeine Beweisidee: Wir implementieren zu H auch ein Entschlüsselungsorakel, und benutzen passives A wie oben.

Erste Variante

Die erste Variante ist aber nicht CCA2-sicher.

Ein OW-CCA2-Angriff geht wie folgt:

- Zu (y,z) wähle d zufällig und erfrage Entschlüsselung m' von $(y, z \oplus d)$.
- Berechne $m = m' \oplus d$. Dann ist m Entschlüsselung von (y,z).

Das Problem ist also, daß man Chiffretexte abändern kann (malleability), und so die Entschlüsselungen legal erfragen kann.

Abhilfe: Plaintext awareness, man soll keine Chiffretexte erzeugen können, wenn man nicht schon den Klartext kennt.

Dies beseitigt auch das Problem der ungültigen Klartexte wie beim Bleichenbacher Angriff.

Zweite Variante

Seien $H, H' : \{0,1\}^* \to \{0,1\}^n$ unabhängige Zufallsorakel.

Verschlüsseln von $m \in \{0,1\}^n$:

- Wähle $r \in \{0,1\}^n$ zufällig.
- Berechne $y = f(r)$ und $z = m \oplus H(r)$.
- Berechne $q = H'(r,m)$.
- Ausgabe des Chiffretexts (y,z,q).

Entschlüsseln von (y,z,q):

- Berechne $r = f^{-1}(y)$ mit der Falltürinformation.
- Berechne $m = z \oplus H(r)$.
- Berechne $q' = H'(r,m)$. Wenn $q' \neq q$, dann Ausgabe ungültig.
- Ausgabe der Nachricht m.

Zweite Variante

Thm: Die zweite Variante ist IND-CCA2 sicher.

Bew: H, H' werden simuliert, wie sie sind. Eine Entschlüsselungsanfrage bzgl. (y,z,q) geht wie folgt vor: Überprüfe in der Definition von H', ob es r,m mit $y = f(r)$, $q = H'(r,m)$ und $m = z \oplus H(r)$ gibt. Wenn nein, dann Antwort „ungültig“. Wenn ja, dann Anwort m (für einen gültigen Chiffretext ist $H'(r,m)$ nur mit insignifikanter Wahrscheinlichkeit nicht definiert, daher kein zu großer Fehler hier).

Gegeben y wählen wir zufälliges z,q,m und wenden A auf m, (y,z,q) wie beim IND-CPA Angriff an. A fragt nicht nach der Entschlüsselung von (y,z,q), sondern muß bei Erfolgswahrscheinlichkeit $1/2+\varepsilon$ wieder $H(r)$ mit Wahrscheinlichkeit $\geq \varepsilon$ erfragen, was uns r liefert. $\square$

Fazit

Beweistechnik für IND-CCA1 und IND-CCA2 Sicherheit:

- CPA Angreifer passiv, löst schwieriges Problem, berechnet also Urbild unter Einwegfunktion f ohne Falltürinformation. Wir müssen nur Hashfunktion simulieren.
- CCA1 und CCA2. Wir müssen Hashfunktion(en) und Entschlüsselungsanfragen simulieren, ohne Falltür von f zu kennen. Damit Reduktion von aktivem auf passiven Angreifer.
- Bedingung an Entschlüsselungsanfragen in CCA1 und CCA2 bewirkt, daß wir mit CPA Angreifer schwieriges Problem mit uns unbekannter Lösung lösen können. Es soll nichts gelöst werden, dessen Lösung wir vorher simuliert haben.