OAEP

OAEP = Optimal Asymmetric Encryption Padding.

Wird in PKCS #1 verwendet. Gutes Verhältnis der Bitlängen von Nachrichten und Chiffretexten (daher „optimal“).

Sei $f : \{0,1\}^k \to \{0,1\}^k$ eine Permutation.

Seien k_0, k_1 mit $k_0 + k_1 < k$ und $2^{-k_0}, 2^{-k_1}$ vernachlässigbar.

Sei $n = k - k_0 - k_1$ und $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$, $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ unabhängige Zufallsorakel.

Verschlüsseln von $m \in \{0,1\}^n$ (Ziel Chiffretext = Ausgabe von f):

- Wähle $r \in \{0,1\}^{k_0}$ zufällig.
- Berechne $y = (m||0^{k_1}) \oplus G(r)$ und $z = r \oplus H(y)$.
- Berechne $c = f(y||z)$. Ausgabe des Chiffretexts c.

OAEP

Entschlüsseln von c:

- Berechne $(y||z) = f^{-1}(c)$ mit der Falltürinformation.
- Berechne $r = z \oplus H(y)$ und $(m||s) = y \oplus G(r)$.
- Wenn $s \neq 0^{k_1}$, dann ungültig. Sonst Ausgabe der Nachricht m.

Betrachte $f : \{0,1\}^k \to \{0,1\}^{n+k_1} \times \{0,1\}^{k_0}$ und die Projektion $\pi : \{0,1\}^{n+k_1} \times \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$.
f ist eine partielle Einwegpermutation, wenn $\pi \circ f$ eine Einwegfunktion ist.

Thm: OAEP ist IND-CCA2 sicher, wenn f eine partielle Einwegpermutation ist. Für beliebige Einwegpermutationen ist dies falsch (sicher nicht unbedingt im praktischen Sinn, siehe auch Diskussion in PKCS #1).

OAEP

Thm: Die RSA Funktion ist eine partielle Einwegpermutation.

Es gibt weitere Padding Verfahren:

- OAEP+
- SEAP, SAEP+

OAEP$^+$ $= ((m \oplus H(r))||W(m,r))||(r \oplus G((m \oplus H(r))||W(m,r)))$.
SAEP$(m,r) = ((m||0^{s_0}) \oplus H(r))||r$.
SAEP$^+(m,r) = ((m||G(m||r)) \oplus H(r))||r$.
$(W, s_0, G, H$ passend).

In OAEP und OAEP mit RSA Modul von 1024 Bits kann man bis 768 Bit Nachrichten sicher verschlüsseln. In SAEP+ nur bis 384 Bits. Sind jeweils IND-CCA2 sicher für Einwegpermutationen mit Falltür.

Gruppenbasierte Kryptographie

Das RSA und Rabin Verfahren basieren auf dem Restklassenring $\mathbb{Z}/n\mathbb{Z}$ und der Einheitengruppe darin.

Wir betrachten nun Kryptosysteme, welche auf beliebigen abelschen Gruppen basieren. Wir nehmen im folgenden an, daß G eine zyklische Gruppe der Ordnung ℓ ist ($\ell = c\ell_0$ mit c klein und ℓ_0 prim).

Sei G erzeugt von g. Für jedes Element $b \in G$ gibt es ein eindeutig bestimmtes $x \in \mathbb{Z}$ mit $0 \leq x < \ell - 1$ und $b = g^x$. Wir nennen x den diskreten Logarithmus von b zur Basis g.

Die Berechnung diskreter Logarithmen in geeigneten Gruppen ist (vermutlich) ein schwieriges Problem (ähnlich wie das Faktorisieren von n). Die Abbildung $x \mapsto g^x$ ist dann eine Einwegfunktion.

Beispiel: $(\mathbb{Z}/p\mathbb{Z}, +)$ leicht, $(\mathbb{Z}/p\mathbb{Z})^\times$ (normalerweise) schwer.

ElGamal Verschlüsselung

Schlüsselerzeugung:
- Wähle $x \in \mathbb{Z}$ mit $0 \leq x < \ell - 1$ zufällig.
- Berechne $y = g^x$.
- Der geheime Schlüssel ist x, der öffentliche Schlüssel ist y.

Verschlüsselung von $m \in G$:
- Wähle $r \in \mathbb{Z}$ zufällig.
- Berechne $u = g^r$ und $v = my^r$.
- Der Chiffretext ist (u, v).

Entschlüsselung von $(u, v) \in G \times G$:
- Berechne $m = vu^{-x}$.
- Der Klartext ist m.

ElGamal Verschlüsselung

Die Abbildung $x \mapsto g^x$ ist keine Einwegfunktion mit Falltür. Dies ist im Verfahren auch nicht erforderlich.

Für G kann man allgemeiner eine Untergruppe der multiplikativen Gruppe von endlichen Körpern $\mathbb{F}_q^\times$ verwenden.

Das ElGamal Verfahren ist randomisiert. Chiffretexte zu zufälligen Nachrichten sind wie zufällige Elemente aus $G \times G$.

Man maskiert eine Nachricht m mit einem zufälligen Wert y^r. Durch die Angabe von g^r versetzt man den Empfänger in die Lage, $y^r = (g^r)^x$ auszurechnen und so m wiederzuerhalten.

ElGamal Sicherheit

Das Diffie-Hellman Problem ist, zu g, g^a, g^b den Wert g^{ab} auszurechnen.

Thm: Das ElGamal Verfahren ist OW-CPA sicher, wenn das Diffie-Hellman Problem schwierig ist.

Bew: Zu g, g^a, g^b wählen wir zufällig $s, r \in \mathbb{Z}$ modulo ℓ und $z \in G$ und wenden einen Angreifer auf g, den öffentlichen Schlüssel $y = g^{as}$ und den Ciffretext (g^{br}, z) an. Wir erhalten $m = zg^{-bras}$, daraus $z/m = g^{absr}$ und schließlich $g^{ab} = (g^{absr})^{1/(sr)}$. $\square$

Bemerkung: Ist $\gcd\{r, \ell\} = 1$, gibt es $\lambda, \mu \in \mathbb{Z}$ mit $1 = \lambda r + \mu\ell$. Damit ist g^λ die eindeutig bestimmte r-te Wurzel von g.

ElGamal Sicherheit

Das Diffie-Hellman Entscheidungsproblem ist, zu g, g^a, g^b, h zu entscheiden, ob $h = g^{ab}$ oder nicht.

Thm: Das ElGamal Verfahren ist IND-CPA sicher, wenn das Diffie-Hellman Entscheidungsproblem schwierig ist.

Bew: Statt einem Angreifer gegen IND betrachten wir einen Angreifer A gegen das Problem A. Zu g, g^a, g^b, h wählen wir ein zufälliges $m \in G$ und wenden A bezüglich des Basiswerts g und des öffentlichen Schlüssels g^a auf m und den „Chiffretext“ (g^b, mh) an. Gilt $h = g^{ab}$, so ist der Chiffretext eine Verschlüsselung von m, ansonsten nicht. $\square$

Drei Probleme

Wir haben also drei Probleme:
- das diskrete Logarithmus Problem (DLP).
- das Diffie-Hellman Problem (CDH, computational DH).
- das Diffie-Hellman Entscheidungsproblem (DDH, decision DH).

Wir können das DDH lösen, wenn wir das CDH lösen können.
Wir können das CDH lösen, wenn wir das DLP lösen können.

Weitere Verhältnisse (grob angedeutet):
- Für allgemeine Gruppen (Black-box Gruppen) ist DDH schwer.
- Es gibt keinen Algorithmus, der das CDH in allgemeinen Gruppen lösen kann, auch unter der Annahme, daß das DDH leicht ist.
- Es gibt Gruppen, in denen das DDH leicht, aber das CDH (vermutlich) schwer ist.
- Es gibt Gruppen, für die das CDH äquivalent zum DLP ist.

ElGamal Sicherheit

Jetzt gibt es wieder (ähnlich wie bei RSA) folgende Fragestellungen:
- Ist ElGamal Verschlüssung IND-CCA2 sicher bzw. was muß man dafür tun (unter der Annahme, daß DDH schwer ist)?
- Für welche speziellen Gruppen ist (oder erscheint) DDH sicher?
- Sind spezielle Bits sicher oder unsicher ...

Wir gehen im folgenden zunächst auf Punkt eins und drei ein, dann fangen wir mit Punkt zwei an.

ElGamal Sicherheit

ElGamal wie bisher vorgestellt wird auch als „Plain ElGamal“ oder „Textbook ElGamal“ bezeichnet.

Plain ElGamal ist nicht Plaintext Aware:
- Jedes Element $(z,h) \in G \times G$ ist die Verschlüsselung der Nachricht $m = hz^{-a}$ unter dem öffentlichen Schüssel g^a.

Plain ElGamal ist nicht NM-CPA sicher:
- Ist (z,h) die Verschlüsselung von m, so ist (z,sh) die Verschlüsselung von sm.

Plain ElGamal ist nicht OW-CCA2 sicher:
- Ein Angreifer erhält die Verschlüsselung (z,h) von m.
- Er erfragt die Entschlüsselung von (z,sh) für ein zufälliges $s \in G$ und erhält m'. Er berechnet $m = m'/s$.

Fujisaki-Okamoto Transformation

Ähnlich wie bei RSA und OAEP kann man durch eine zusätzliche Konstruktion ein IND-CCA2 sicheres Kryptosystem bekommen.

Die Konstruktion heißt Fujisaki-Okamoto Transformation (2000).

Seien $\mathcal{E}$ und $\mathcal{D}$ Ver- und Entschlüsselungsfunktionen.
- Die Verschlüsselung von Nachrichten m unter dem öffentlichen Schlüssel y sei $c = \mathcal{E}_y(m,r)$, wobei r den in der Verschlüsselung benutzten zufälligen Wert bezeichnet.
- Die Entschlüsselung sei $m = \mathcal{D}_a(c)$, wobei a den geheimen Schlüssel bezeichnet.

Sei H eine Hashfunktion (mit passendem Bildbereich) im Zufallsorakelmodell.

Fujisaki-Okamoto Transformation

Fujisaki-Okamoto transformierte Verschlüsselung $\bar{\mathcal{E}}_y(m,r)$:

- $c = \bar{\mathcal{E}}_y(m,r) = \mathcal{E}_y((m||r), H(m||r))$.

Fujisaki-Okamoto transformierte Entschlüsselung $\bar{\mathcal{D}}_a(c)$:

- Berechne $(m||r) = \mathcal{D}_a(c)$.
- Wenn $\mathcal{D}_a(c)$ Fehler ergibt, dann Ausgabe Fehler. Wenn $c \neq \mathcal{E}_y((m||r), H(m||r))$, dann Ausgabe Fehler.
- Ansonsten Ausgabe m.

Thm (RO): Wenn $(\mathcal{E}, \mathcal{D})$ IND-CPA sicher ist, dann ist $(\bar{\mathcal{E}}, \bar{\mathcal{D}})$ IND-CCA2 sicher.

Bit Sicherheit

Thm: Ist $\gcd\{d,\ell\} = 1$ und kann man aus g, g^a den Wert $a \bmod d$ berechnen, so kann man auch ganz a berechnen.

Bew: Der Wert $g^a g^{-(a \bmod d)}$ hat einen durch d teilbaren Exponenten. Die $\gcd$-Bedingung sagt, daß wir eindeutig d-te Wurzeln ziehen können. Wir können daher $g^{a \text{ div } d}$ berechnen. Induktiv erhalten wir schließlich $g^{a \text{ div } d} = 1$ und haben a zur Basis d mit nicht negativen Koeffizienten dargestellt. $\square$

Für $d = 2$ ergibt sich die Äquivalenz der Sicherheit des niedrigsten Bits von a mit der Sicherheit von ganz a.

Beispiel: Ist $G = \mathbb{F}_p^\times$, so ist $\ell = p-1$ gerade. Durch Exponieren mit $(p-1)/2$ bilden wir jedes $z \in G$ auf $\{-1, 1\}$ ab (Jacobisymbol). Für den Wert g^a gilt $a \bmod 2 = 0$ genau dann, wenn $a^{(p-1)/2} = 1$. Wir können jedoch keine eindeutigen Quadratwurzeln ziehen ...

Bit Sicherheit

Beispiel (ctd.): Sei G' die Untergruppe von G der Ordnung $(p-1)/2$, erzeugt von g^2. Hierin ist das 0-te Bit wieder sicher, wenn $(p-1)/2$ ungerade ist und das DLP sicher ist. Das 0-te Bit in G' bezüglich g^2 entspricht dem 1-ten Bit in G bezüglich g, welches daher ebenfalls sicher ist.

Bit Sicherheit in spezieller Gruppe G:

Thm: Sei $G = \mathbb{F}_p^\times$, $n = \log_2(p)$ und $\varepsilon > 0$. Können wir aus g, g^a, g^b die $\varepsilon\sqrt{n}$ höchsten Bits von g^{ab} effizient (in Abhängigkeit von n) berechnen, so können wir ganz g^{ab} effizient berechnen. Die Laufzeit ist exponentiell in $1/\varepsilon$.

Untere Schranken für das DDH

Für konkret gegebene Gruppen gibt es keine (sinnvollen) unteren Schranken für die Laufzeit, ein DLP zu lösen. Für „allgemeine“ Gruppen gibt es dies aber.

Eine Numeration ist eine Injektion $\sigma : (\mathbb{Z}/\ell\mathbb{Z})^+ \to \{0,1\}^n$.

Ein generischer Algorithmus erwartet als Eingabe eine Liste von Gruppenelemente $L = (\sigma(x_1), \ldots, \sigma(x_k))$ und hat Zugang zu einem Orakel, welches die Werte $\sigma(x_i \pm x_j)$ berechnet. Solche neu berechneten Werte werden zu L hinzugefügt.

Alternativ kann man sich vorstellen, die Gruppe sei durch eine Klasse gegeben, und der generische Algorithmus kann nur die Methoden $+, -, =$ aufrufen (sonst nichts).

Untere Schranken für das DDH

Thm: Sei ℓ prim. Das DDH kann durch einen generischen Algorithmus A bei zufälliger Numeration σ und $\leq m$ Orakelanfragen an σ mit von $1/2$ um maximal m^2/ℓ abweichender Wahrscheinlichkeit gelöst werden.

Bew: A erhält $\sigma(1), \sigma(a), \sigma(b), w_s, w_{1-s}$ mit $w_0 = \sigma(ab)$, $w_1 = \sigma(c)$, $s \in \{0,1\}$ und $a,b,c \in \mathbb{Z}/\ell\mathbb{Z}$ zufällig. Die durch A berechneten Elemente sind von der Form $\sigma(F_j(a,b,c))$ für $F_j(t_1,t_2,t_3) = \lambda_{j,1}t_1 + \lambda_{j,2}t_2 + \lambda_{j,3}t_3 + \lambda_{j,4}t_1t_2$ und $\lambda_{j,i} \in \mathbb{Z}/\ell\mathbb{Z}$. A kann nur dann Information erhalten, wenn $\sigma(F_i(a,b,c)) = \sigma(F_j(a,b,c))$ für i,j mit $F_i \neq F_j$. Tritt dies nicht ein, hat A Erfolgswahrscheinlichkeit $1/2$. Wir suchen also nach der Wahrscheinlichkeit, daß ein $G_{i,j} = F_i - F_j$ eine Nullstelle (a,b,c) hat. Da immer nach einem t_i aufgelöst werden kann, nachdem Werte für die anderen t_j eingesetzt wurden, ist $G_{i,j}$ für zufällige Wahl von t_i mit Wahrscheinlichkeit $1/\ell$ Null. Es gibt $m(m-1)/2$ Polynome $G_{i,j}$, somit eine Wahrscheinlichkeit $\leq m(m-1)/(2\ell) \leq m^2/\ell$. $\square$

Untere Schranken für das DDH

Der Satz gilt analog für das DLP, Erfolgswahrscheinlichkeit $\leq m^2/\ell$.

Um konstante Erfolgswahrscheinlichkeit zu haben, benötigt man also $m =$ Omega($\sqrt{\ell}$) viele Orakelanfragen. Die Laufzeit ist demnach exponentiell in $\log(\ell)$.

Mit den Pollard Methoden ergibt sich, daß $O(\sqrt{\ell})$ auch eine obere Schranke für die benötigte Zeit ist, folglich ist Theta($\sqrt{\ell}$) die genaue Komplexität für generische Algorithmen bzw. Black-box Gruppen.