Gruppenbasierte Kryptographie

Benutzt zyklische Gruppen von (fast) Primzahlordnung:
$G = \langle g \rangle$ und $\#G = \ell = c\ell_0$ mit c klein und ℓ_0 prim.

$\forall b \in G : \exists x \in \mathbb{Z} : b = g^x$. Das Element x heißt diskreter Logarithmus von b zur Basis g, ist modulo ℓ eindeutig bestimmt.
Diskretes Logarithmus Problem (DLP): Gegeben g, b finde x.

Für geeignete Gruppen ist das DLP (vermutlich) schwer, also $x \mapsto g^x$ eine Einwegfunktion.

Mit zyklischen Gruppen kann man machen:

- ElGamal Verschlüsselung (ca. 1985)
- ElGamal, Schnorr, etc. Unterschriften (ca. 1985-1991)
- Diffie-Hellman Schlüsselaustausch (ca. 1976).
- vieles weitere ...

ElGamal Verschlüsselung

Schlüsselerzeugung:

- Wähle $x \in \mathbb{Z}$ mit $0 \le x < \ell - 1$ 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 Sicherheit

Das Diffie-Hellman Problem (CDH) ist, für g, g^a, g^b den Wert g^{ab} auszurechnen.

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

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

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

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.

ElGamal Sicherheit

Bereits betrachtet:

1. Plain ElGamal nicht NM-CPA oder OW-CCA2 sicher. Auch nicht PA. Daher Fujisaki-Okamoto Transformation verwenden. Macht aus IND-CPA sicherem Verschlüsselungsverfahren ein IND-CCA2 sicheres Verfahren (im Zufallsorakelmodell).

2. Vorsicht, daß wirklich IND-CPA sicher (Jacobi Symbol in $\mathbb{F}_p$ ...).

Nun:

3. Das DLP, CDH und DDH in allgemeinen Gruppen betrachten.

4. Konkrete Gruppen betrachten.

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. □

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 $m = \text{Omega}(\sqrt{\ell})$ 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 $\text{Theta}(\sqrt{\ell})$ die genaue Komplexität für generische Algorithmen bzw. Black-box Gruppen.

Shanks Baby Step Giant Step

Ist Anwendung von Meet-in-the-Middle Prinzip. Löst DLP deterministisch in Zeit $O(\sqrt{\ell})$ und Speicher $O(\sqrt{\ell})$.

Schreibe x mit $0 \leq x < \ell$ eindeutig als $x = jm + i$ mit $0 \leq i < m$ und $0 \leq j \leq \ell/m$. Schreibe $x \mapsto g^x$ als $(i, j) \mapsto g^{jm+i}$.

Seien g, b gegeben und x mit $b = g^x$ gesucht.
Trenne i und j: Für $x = jm + i$ gilt $b/g^i = g^{jm}$.

Daher linke Seite für alle i ausrechnen und speichern, dann alle j für Kollision durchprobieren.

Zeit-optimiert für $m \approx \sqrt{\ell}$.

Pollard rho

Analog wie beim Kollisionsfinden für Hashfunktionen. Löst DLP probabilistisch in erwarteter Zeit $O(\sqrt{\ell})$ und Speicher $O(1)$ (im RO).

Idee: Wir brauchen einen Zufallsweg (random walk) in G von der Form $b^{u_i}g^{v_i}$ mit bekannten u_i, v_i, welche nur von $b^{u_{i-1}}g^{v_{i-1}}$ abhängen. Eine Kollision $b^{u_i}g^{v_i} = b^{u_j}g^{v_j}$ liefert dann $b^{u_i - u_j} = g^{v_j - v_i}$ und folglich $x = (u_i - u_j)/(v_j - v_i) \bmod \ell$, wenn $(v_j - v_i) \neq 0 \bmod \ell$.

Definition von u_i, v_i ($u_0 = 0$, $v_0 = 0$):

- durch $u_i = H(b^{u_{i-1}}g^{v_{i-1}}||0)$ und $v_i = H(b^{u_{i-1}}g^{v_{i-1}}||1)$ für eine Hashfunktion H.
- oder wie folgt: Zerlege G in disjunkte Teilmengen S_1, S_2, S_3 und

  definiere $(u_i, v_i) = \begin{cases} (u_{i-1}, v_{i-1}+1) & \text{für } b^{u_{i-1}}g^{v_{i-1}} \in S_1 \\ (u_{i-1}+1, v_{i-1}) & \text{für } b^{u_{i-1}}g^{v_{i-1}} \in S_2 \\ (2u_{i-1}, 2v_{i-1}) & \text{für } b^{u_{i-1}}g^{v_{i-1}} \in S_3 \end{cases}$.

Pohlig-Hellman

Beruht auf dem Struktursatz für endlich erzeugte, abelsche Gruppen.

Löst DLP für G beliebig zyklisch, ℓ_0 größter Primfaktor von $\#G$, durch Shanks oder Pollard Methoden in Laufzeit $O^{\sim}(\sqrt{\ell_0})$ und Speicher $O(\sqrt{\ell_0})$ oder $O(1)$.

Idee 1 (chinesischer Restsatz):

- Sei $\#G = \prod_{i=0}^{r} \ell_i^{e_i}$, $n_i = 1 \bmod \ell_i^{e_i}$ und $n_i = 0 \bmod \ell_j^{e_j}$ für alle $j \neq i$, $\phi_i : G \to G$ mit $\phi(z) = z^{n_i}$. Wegen $\gcd\{n_0, \ldots, n_r\} = 1$ gilt $\cap_i \ker(\phi_i) = 1$.
- Die Ordnung von $G_i = \phi_i(G)$ ist $\ell_i^{e_i}$.
- DLP in jedem G_i lösen (falls lösbar) und mit chinesischem Restsatz zusammensetzen: Finde x_i mit $\phi_i(b) = \phi_i(g)^{x_i}$. Dann ist $x = \sum_i x_i n_i$ der DL.

Pohlig-Hellman

Idee 2 („Hensel Lifting“):

- Annahme $\#G = \ell^e$, ℓ prim und $g^{\ell^e} = 1$ mit e minimal.
- Dann $b = g^x$ mit $x = x_0 + x_1\ell + \ldots x_{e-1}\ell^{e-1}$ und $0 \leq x_i < \ell$.
- Löse induktiv DLPs $b^{\ell^{e-1-j}} / g^{\ell^{e-1-j}(x_0 + x_1\ell + \ldots x_{j-1}\ell^{j-1})} = g^{\ell^{e-1-j} x_j \ell^j}$ in x_j für $j = 0, \ldots, e-1$. Sind definiert in Gruppe $\langle g^{\ell^{e-1}} \rangle$ der Ordnung ℓ.

Daher betrachten wir nur zyklische Gruppen mit „möglichst“ primer Ordnung.
Die Verwendung nicht zyklischer Gruppen macht ebenfalls keinen Sinn, da wir nur in der von g erzeugten Untergruppe arbeiten.