VI.4 Elgamal

- vorgestellt 1985 von Taher Elgamal
- nach RSA das wichtigste Public-Key Verfahren
- besitzt viele unterschiedliche Varianten, abhängig von zugrunde liegender zyklischer Gruppe
- Elgamal auf Gruppen definiert über elliptischen Kurven gewinnt immer größere Bedeutung
- betrachten nur den einfachsten Fall
- Verallgemeinerung problemlos

Vorbereitungen

Definition 6.17 Eine endliche Gruppe G heißt zyklisch, wenn es ein $g \in G$ gibt mit $G = \left\{ g^i : 0 \le i \le |G| - 1 \right\}$. Das Element g heißt dann erzeugendes Element oder Generator von G.

Satz 6.18 Sei p eine Primzahl. Dann ist die Gruppe $\mathbb{Z}_p^*$ zyklisch.

Zur Erinnerung Ist p Primzahl, so gilt $\left|\mathbb{Z}_p^*\right| = p - 1$.

Beispiel Sei $p = 7$, dann ist $g = 3$ ein Generator der Gruppe $\mathbb{Z}_7^*$. $g = 5$ ist ebenfalls ein Generator von $\mathbb{Z}_7^*$.

Elgamal – Schlüsselerzeugung

1. Erzeuge eine Primzahl p und einen Generator g der Gruppe $\mathbb{Z}_p^*$.
2. Wähle ein zufälliges Element $a \in \{0,1,\ldots,p-2\}$ und setze $h := g^a \mod p$.
3. Der öffentliche Schlüssel ist $pk := (p,g,h)$, der private Schlüssel ist $sk := (p,g,a)$.

Elgamal – Verschlüsselung

Bei Wahl des öffentlichen Schlüssels $pk := (p, g, h)$ ist der Klartextraum $P = \mathbb{Z}_p$

Der Chiffretextraum ist $C := \mathbb{Z}_p^* \times \mathbb{Z}_p$

Verschlüsselung bei Klartext $m \in \mathbb{Z}_p$

1. Wähle $r \in \{0, 1, \ldots, p-2\}$ zufällig.
2. Setze $v := g^r \bmod p$ und $w := h^r \cdot m \bmod p$.
3. Der Chiffretext ist $c := (v, w)$.

Verschlüsselung in Elgamal ist keine Funktion, sondern ein randomisierter Algorithmus!

Elgamal – Entschlüsselung

Bei Wahl des öffentlichen Schlüssels $pk := (p, g, h)$ ist der Klartextraum $P = \mathbb{Z}_p$.

Der Chiffretextraum ist $C := \mathbb{Z}_p^* \times \mathbb{Z}_p$.

Entschlüsselung bei Chiffretext $c = (v, w) \in \mathbb{Z}_p^* \times \mathbb{Z}_p$

1. Setze $u := v^a \mod p$ und $m := u^{-1} \cdot w \mod p$.
2. Der Klartext ist m.

Korrektheit $v^a = (g^r)^a = g^{ar} = (g^a)^r = h^r \mod p$

Effizienz von Ver- und Entschlüsselung

Lemma 6.19 Sind der öfffentliche und geheime Schlüssel (p,g,h) und (p,g,a), so kann die Verschlüsselung eines Klartexts und die Entschlüsselung eines Chiffretexts in Zeit $\mathcal{O}\left(\log(p)^3\right)$ berechnet werden.

Square-and-Multiply

Ziel G endliche Gruppe, $g \in G$, $a \in N$, $a = \sum_{i=0}^{l-1} a_i 2^i$, $a_i \in \{0,1\}$,

berechne g^a in G.

```
Square-and-Multiply(g,a)
1  y := 1
2  z := g
3  for i = 0 to l – 1 do
4      if a_i = 1
5          then y := y · z
6      z := z^2
7  return y
```

Elgamal - Schlüsselerzeugung

Zu zeigen sind 2 Dinge:

1. **Primzahl p kann effizient erzeugt werden.**

2. **Generator g kann effizient erzeugt werden.**

Primzahlerzeugung

Beruht auf zwei Tatsachen:

1. **Es kann effizient entschieden werden, ob eine Zahl Primzahl ist (z.B. Miller-Rabin-Test).**
2. **Es gibt “viele” Primzahlen (Primzahlsatz).**

Dann: Erzeuge zufällige Zahlen und teste, ob sie Primzahlen liefern.

Erzeugung von Generatoren

1. Wählen Primzahl q, so dass $p = 2q + 1$ ebenfalls Primzahl ist (Sophie Germain Primzahlen).
2. $\mathbb{Z}_p^*$ besitzt dann $q - 1$ Generatoren.
3. $g \in \mathbb{Z}_p^*$ genau dann Generator, wenn $g^2 \neq 1 \mod p$ und $g^q \neq 1 \mod p$.

Erzeugung eines Generators mod p

Erzeuge zufällige Elemente aus $\mathbb{Z}_p^*$ und teste, ob sie Generator sind, bis ein Generator gefunden.

Beobachtung Falls $p = 2q + 1$ für eine Primzahl q, so besitzt Z_p^* $q - 1$ Generatoren.

Sicherheit des geheimen Schlüssels

Diskrete Lograithmen Sei G eine zyklische Gruppe und g ein Generator von G. Ist $h \in G$, so bezeichnen wir mit $\mathrm{dlog}_g(h)$ die eindeutige Zahl $a \in \{0,1,\ldots,|G|-1\}$ mit $g^a = h$.

Private Schlüssel und diskrete Logarithmen Soll bei der Elgamal-Verschlüsselung ein geheimer $sk = (p,g,a)$ aus dem zugehörigen öffentlichen Schlüssel $pk = (p,g,h)$ berechnet werden, so ist dies äquivalent zur Berechnung des diskreten Logarithmus von h zum Generator g in $\mathbb{Z}_p^*$.

Diskrete Logarithmen

Tatsache 6.20 Der zurzeit beste bekannte Algorithmus zur Berechnung des diskreten Logarithmus in den Gruppen $\mathbb{Z}_p^*$ besitzt Laufzeit $e^{c\cdot\log(p)^{1/3}\cdot\log\log(p)^{2/3}}$, wobei $c \approx 1.94$.

Satz 6.21 Sei p eine Primzahl und sei $p-1=\prod_{i=1}^{k} q_i^{e_i}$ die Primfaktorzerlegung von $p-1$. Dann können diskrete Logarithmen in Z_p^* in Zeit $O\left(\log(p)^3 \cdot \max\{q_j\}\right)$ berechnet werden.

Angriffe auf Elgamal und das DH-Problem

Diffie-Hellman (DH) Problem Sei G eine zyklische Gruppe und g ein Generator von G. Beim Diffie-Hellmann Problem muss aus den Gruppenelementen g^a und g^b das Gruppenelement $g^{a \cdot b}$ berechnet werden.

Entschlüsselung und das DH-Problem Soll bei der Elgamal-Verschlüsselung ohne Kenntnis des privaten Schlüssels $sk = (p, g, a)$ aus einem Chiffretext $c = (v, w)$ der zugehörige Klartext m berechnet werden, so ist dies äquivalent zur Lösung des DH-Problems in $\mathbb{Z}_p^*$.

Chosen-Ciphertext Angriffe auf Elgamal

Satz 6.22 **Elgamal ist nicht sicher gegen Chosen-Ciphertext Angriffe.**

Möglichkeiten eines Angreifers

- **Ciphertext-Only Angriff** Angreifer kennt nur Chiffretext c.
- **Known-Plaintext Angriff** Angreifer kennt Chiffretext c und Paare (m_i,c_i) von Klartexten und Chiffretexten unter dem gleichen Schlüssel e.
- **Chosen-Plaintext Angriff** Angreifer kann sich zu selbst gewählten Klartexten m_i die Chiffretexte c_i erzeugen.
- **Chosen-Ciphertext Angriff** Angreifer kann sich zu selbst gewählten Chiffretexten c_i die Klartexte m_i erzeugen.

V.5 Hybrid -Verfahren

- Public-Key Verfahren deutlich ineffizienter als symmetrische Verfahren.

- Daher ungeeignet zur Verschlüsselung großer Daten.

- Zur Verschlüsselung großer Daten kann eine Kombination aus Public-Key und symmetrischer Verschlüsselung benutzt werden (Hybrid-Verfahren).

Hybrid-Verfahren

$V^1 = (P^1, C^1, K^1, E^1, D^1)$ symmetrisches Verfahren

$V^2 = (P^2, C^2, K^2, E^2, D^2)$ Public-Key Verfahren mit $K^1 \subseteq P^2$,

d.h. mit V^2 können Schlüssel von V^1 verschlüsselt werden.

Verschlüsselung einer Nachricht m von B nach A

1. B wählt $k \in K^1$.
2. B berechnet Chiffretext c als (v, w) mit

   $v = E^2_{pk_A}(k)$ und $w = E^1_k(m)$.

Dabei ist pk_A der öffentliche Schlüssel von A in V^2.

Hybrid-Verfahren

Verschlüsselung einer Nachricht m von B nach A

1. B wählt $k \in K^1$.
2. B berechnet Chiffretext c als (v, w) mit

   $v = E^2_{pk_A}(k)$ und $w = E^1_k(m)$.

Dabei ist pk_A der öffentliche Schlüssel von A in V^2.

Entschlüsselung eines Chiffretextes $c = (v, w)$ durch A

1. A berechnet $k = D^2_{sk_A}(v)$.
2. A berechnet $m = D^1_k(w)$.

Dabei ist sk_A der private Schlüssel von A in V^2.

V.6 Schlüsselaustausch und das DH-Protokoll

Ein Protokoll ist ein Algorithmus zwischen mehreren Teilnehmern, um ein spezifisches Ziel zu erreichen. Ein Protokoll ist definiert durch eine Folge von Aktionen der Teilnehmer.

Aktionen sind Berechnungen eines Teilnehmers oder das Senden einer Nachricht.

Ein interaktives Schlüsselaustauschprotokoll ist ein Protokoll, durch das ein gemeinsamer geheimer Schlüssel für die Teilnehmer als Funktion der ausgetauschten Informationen erzeugt wird, ohne dass ein Teilnehmer das Resultat vorhersagen kann.

Angreifer und Sicherheit

Ein passiver Angreifer versucht den geheimen Schlüssel nur durch Analyse der gesendeten Nachrichten zu bestimmen.

Ein aktiver Angreifer verfälscht Nachrichten und injiziert eigene Nachrichten (unter falscher Identität).

Ein interaktives Schlüsselaustauschprotokoll ist sicher gegen passive Angriffe, wenn kein passiver Angreifer mit vertretbaren Aufwand und mit signifikanter Wahrscheinlichkeit gemeinsame geheime Schlüssel bestimmen kann.

Das Diffie-Hellman Protokoll

Definition 6.23 Sei p eine Primzahl. Das Diffie-Hellman (DH) Protokoll über $\mathbb{Z}_p^*$ ist definiert durch die folgenden Aktionen.

1. A wählt $x \in \{0,1,\ldots,p-1\}$ zufällig gleichverteilt.
2. A berechnet $h_1 := g^x \mod p$ und sendet h_1 an B.
3. B wählt $y \in \{0,1,\ldots,p-1\}$ zufällig gleichverteilt.
4. B berechnet $h_2 := g^y \mod p$ und sendet h_2 an A.
5. A setzt ihren gemeinsamen geheimen Schlüssel auf $k_A := h_2^x \mod p$.
6. B setzt seinen gemeinsamen geheimen Schlüssel auf $k_B := h_1^y \mod p$.

Das Diffie-Hellman Protokoll

A

B

$x \leftarrow_R \mathbb{Z}_{p-1}$

$h_1 := g^x \bmod p$

h_1 →

$y \leftarrow_R \mathbb{Z}_{p-1}$

$h_2 := g^y \bmod p$

← h_2

$k_A := h_2^x \bmod p$

$k_B := h_1^y \bmod p$

Das Diffie-Hellman Protokoll

A

B

$x \leftarrow_R \mathbb{Z}_{p-1}$

$h_1 := g^x \bmod p$

h_1 →

← h_2

$y \leftarrow_R \mathbb{Z}_{p-1}$

$h_2 := g^y \bmod p$

$k_A := h_2^x \bmod p$

$k_B := h_1^y \bmod p$

Korrektheit $k_B = h_1^y = \left(g^x\right)^y = g^{xy} = \left(g^y\right)^x = h_2^x = k_A \bmod p$

Das Diffie-Hellman Protokoll

A				B
$x \leftarrow_R \mathbb{Z}_{p-1}$				$y \leftarrow_R \mathbb{Z}_{p-1}$
$h_1 := g^x \bmod p$		h_1		$h_2 := g^y \bmod p$
		$\longrightarrow$		
		h_2		
		$\longleftarrow$		
$k_A := h_2^y \bmod p$				$k_B := h_2^x \bmod p$

Lemma 6.24 Jeder Teilnehmer im DH-Protokoll kann seine Berechnungen in Zeit $O\left(\log(p)^3\right)$ ausführen.

Sicherheit gegen passive Angreifer

Diffie-Hellman (DH) Problem Sei G eine zyklische Gruppe und g ein Generator von G. Beim Diffie-Hellmann Problem muss aus den Gruppenelementen g^a und g^b das Gruppenelement $g^{a \cdot b}$ berechnet werden.

Passive Angreifer und DH-Problem Um erfolgreich zu sein, muss ein passiver Angreifer das DH-Problem in der Gruppe $\mathbb{Z}_p^*$ lösen. Umgekehrt, liefert ein Algorithmus für das DH-Problem in $\mathbb{Z}_p^*$ einen passiven Angreifer gegen das DH-Protokoll.

Bemerkungen

- Das DH Protokoll ist nicht sicher gegen aktive Angriffe.
- Um Sicherheit gegen aktive Angriffe zu erhalten, wird Authentizität benötigt.
- Um gemeinsame geheime Schlüssel in $\{0,1\}^*$ zu erhalten, werden Hashfunktionen und Zufallsextraktoren benötigt.