Broadcast encryption with traitor tracing

Chi-Thanh Christopher Nguyen <chithanh@cs.tu-berlin.de>

4. Januar 2006

Seminar Algorithmische Algebra und Zahlentheorie, TU Berlin

Dozenten: Prof. Dr. M. Pohst und Prof. Dr. F. Heß

Übersicht

In diesem Vortrag wird ein Verfahren für die broadcast encryption vorgestellt, dass einen Verräter (traitor) identifizieren kann, dessen Schlüssel kompromittiert ist [1]. Schließlich werden mögliche Gründe diskutiert, warum diese Verfahren derzeit nicht zum Einsatz kommen.

Traitor tracing

Ein traitor tracing Verfahren besitzt idealerweise folgende Eigenschaften:

- *public traceability*, d.h. die Information, die zum Identifizieren eines Verräters notwendig ist, sollte nicht das Entschlüsseln der Nachricht ermöglichen.
- *black-box traceability*, d.h. die Zuordnung eines Decoders zu einem Verräter soll ohne Kenntnis des enthaltenen geheimen Schlüssels möglich sein.
- *collusion resistance*, d.h. ein Verräter soll auch dann identifiziert werden, wenn eine vorgegebene Anzahl Verräter untereinander kommunizieren können.
- *constant transmission rate*, d.h. der Chiffretext wächst asymptotisch höchstens um einen konstanten Faktor.

Wiederholung Diffie-Hellman Problem

Seien G_1 Gruppe von Primzahlordnung q, und P Erzeuger von G_1.

- CDH (computational Diffie-Hellman)
  Gegeben (P, aP, bP) für $a, b \in \mathbb{Z}_q^\times$, berechne abP

- CBDH[1] (computational bilinear Diffie-Hellman)
  Gegeben (P, aP, bP, cP) für $a, b, c \in \mathbb{Z}_q^\times$, berechne $abcP$

- DBDH[1] (decisional bilinear Diffie-Hellman)
  Gegeben (P, aP, bP, cP, U) für $a, b, c \in \mathbb{Z}_q^\times$ und $U \in G_1$, entscheide ob $U = abcP$

Für den Fall $b = c$ erhalten wir folgende modifizierte Probleme:

- CBDH^1-M (modified computational bilinear Diffie-Hellman)
  Gegeben (P, aP, bP) für $a, b \in \mathbb{Z}_q^\times$, berechne ab^2P

- DBDH^1-M (modified decisional bilinear Diffie-Hellman)
  Gegeben (P, aP, bP, U) für $a, b \in \mathbb{Z}_q^\times$ und $U \in G_1$, entscheide ob $U = ab^2P$

Pairing-basierte Probleme

Sei G_2 ebenfalls Gruppe von Primzahlordnung q. Eine Abbildung $\hat{e} : G_1 \times G_1 \to G_2$ heiße zulässig (admissible), wenn sie bilinear, also $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ für alle $a, b \in \mathbb{Z}_q^\times$ und $P, Q \in G_1$, nicht degeneriert und effizient zu berechnen ist. $\hat{e}$ kann beispielsweise mittels Tate-Pairing konstruiert werden.

- CBDH^2 (computational bilinear Diffie-Hellman)
  Gegeben (P, aP, bP, cP) für $a, b, c \in \mathbb{Z}_q^\times$, berechne g^{abc}, wobei $g = \hat{e}(P, P)$.

- DBDH^2 (decisional bilinear Diffie-Hellman)
  Gegeben (P, aP, bP, cP, Z) für $a, b, c \in \mathbb{Z}_q^\times$ und $Z \in G_2$, entscheide ob $Z = g^{abc}$, wobei $g = \hat{e}(P, P)$.

- CBDH^2-E (extended computational bilinear Diffie-Hellman)
  Gegeben (P, aP, bP, cP, ab^2P) für $a, b, c \in \mathbb{Z}_q^\times$, berechne g^{cb^2}, wobei $g = \hat{e}(P, P)$.

- DBDH^2-E (extended decisional bilinear Diffie-Hellman)
  Gegeben $(P, aP, bP, cP, ab^2P, Z)$ für $a, b, c \in \mathbb{Z}_q^\times$ und $Z \in G_2$, entscheide ob $Z = g^{cb^2}$, wobei $g = \hat{e}(P, P)$.

  Eine Variante von CBDH^2:

- CBDH^2-V (variant computational bilinear Diffie-Hellman)
  Gegeben $(P, aP, bP, cP, a(a^2 - b^2)P, b(a^2 - b^2)P)$ für $a, b, c \in \mathbb{Z}_q^\times$, berechne g^{abc}, wobei $g = \hat{e}(P, P)$.

Mischprobleme

- MCDH (mixed computational Diffie-Hellman)
  Gegeben (P, aP, a^2P, g^b) für $a, b \in \mathbb{Z}_q^\times$, berechne g^{ba^2}, wobei $g = \hat{e}(P, P)$.

- MDDH (mixed decisional Diffie-Hellman)
  Gegeben (P, aP, a^2P, g^b, Z) für $a, b \in \mathbb{Z}_q^\times$ und $Z \in G_2$, entscheide ob $Z = g^{ba^2}$, wobei $g = \hat{e}(P, P)$.

2-user traitor tracing

Setup:

- Eingabe: Security parameter $\kappa \in \mathbb{Z}$, Nachrichtenraum $M = \{0,1\}^\kappa$
- Erzeuge κ-bit Primzahl q, zwei Gruppen G_1, G_2 der Ordnung q, und eine zulässige bilineare Abbildung $\hat{e} : G_1 \times G_1 \to G_2$. Wähle einen Erzeuger $P \in G_1$ und setze $g := \hat{e}(P,P)$ (Anmerkung: g ist Erzeuger von G_2).
- Wähle zufällig $a, z \in \mathbb{Z}_q^\times$ und setze $Q := aP, Z := g^z$.
- Wähle eine Hashfunktion $H : G_1 \to M$.

Systemparameter sind $(q, G_1, G_2, \hat{e}, P, H)$.

Private key: das Paar (a, z)

Encryption key: $pk := (g, Q, Z)$

User key: Benutzer u_i mit $i \in \{0, 1\}$ wird zugeordnet (α_i, β_i) mit $\alpha_i + a\beta_i = z \mod q$

Proxy Quantity: Der Benutzer erhält die Proxy Quantity $\Pi = (\alpha_1, \beta_1 P)$

Encrypt: Erzeuge $k \in \mathbb{Z}_q$ zufällig. $C = (c_1, c_2, d) = (kP, k^2Q, m \oplus H(Z^{k^2}))$. (c_1, c_2, d) ist gültiger Chiffretext, wenn $k \in \mathbb{Z}_q$ existiert so dass $c_1 = kP, c_2 = k^2Q$.

Decrypt: Berechne $Z^{k^2} = \hat{e}(\alpha_i c_1, c_1) \cdot \hat{e}(\beta_i P, c_2)$ und $m = d \oplus H(Z^{k^2}))$.

Decrypt

Korrektheit:

$$
\begin{aligned}
& \hat{e}(\alpha c_1, c_1) \cdot \hat{e}(\beta P, c_2) \\
&= \hat{e}(\alpha k P, kP) \cdot \hat{e}(\beta P, k^2 a P) \\
&= \hat{e}(P,P)^{\alpha k^2} \cdot \hat{e}(P,P)^{a\beta k^2} \\
&= g^{(\alpha + a\beta)k^2} \\
&= (g^z)^{k^2} \\
&= Z^{k^2}.
\end{aligned}
$$

Sicherheit: Im Zufallsorakelmodell ist die Verschlüsselung IND-CPA sicher unter MCDH. Ist H eine universelle Hashfunktion, dann ist sie IND-CPA sicher unter MDDH.

Public black-box traitor tracing

- Wähle $k, k' \in \mathbb{Z}_q^\times$ zufällig.
- Definiere $u_0 := \alpha_0 k^2 + a\beta_0 k'^2$ und $u_1 := \alpha_1 k^2 + a\beta_1 k'^2$
- Berechne $g^{u_0} = \hat{e}(a_0 P, k^2 P) \cdot \hat{e}(Q, k'^2(\beta_0 P))$ und $g^{u_1} = \hat{e}(a_1 P, k^2 P) \cdot \hat{e}(Q, k'^2(\beta_1 P))$.
- Lasse den Piraten-Decoder folgendes entschlüsseln: $(kP, ak'^2 P, d)$.
- Gibt der Decoder d/g^{u_0} aus, ist Benutzer 0 der Verräter, gibt er d/g^{u_1} aus, ist Benutzer 1 der Verräter. Ansonsten sind beide Verräter.

Es gilt:

$$\begin{aligned} & \hat{e}(\alpha kP, kP) \cdot \hat{e}(\beta P, ak'^2 P) \\ = & \hat{e}(P,P)^{\alpha k^2} \cdot \hat{e}(P,P)^{a\beta k'^2} \\ = & \hat{e}(\alpha P, k^2 P) \cdot \hat{e}(aP, k'^2 \beta P) \\ = & \hat{e}(\alpha P, k^2 P) \cdot \hat{e}(Q, k'^2(\beta P)) \end{aligned}$$

Notwendig zum Identifizieren des Verräters ist also lediglich $(\alpha P, \beta P)$. Insbesondere bleibt der Private key geheim, und es wird keine Information zum Bau eines Decoders preisgegeben.

Sicherheit bei öffentlicher traitor tracing Information: IND-CPA unter CBDH^2-E bzw. DBDH^2-E

N-user c-traitor tracing

Das Verfahren lässt sich von zwei auf beliebig viele Nutzer erweitern:

Gegeben ein Code $C = \{\omega_1, \ldots, \omega_N\}$ über dem Alphabet $\{0, 1\}$, der (N, c, l, ϵ)-sicher gegen collusion ist. Dieser ermöglicht es uns, l 2-user-Systeme zu kombinieren.

Setup:

- Eingabe: Security parameter κ, c, ϵ.
- Erzeuge κ-bit Primzahl q, zwei Gruppen G_1, G_2 der Ordnung q und eine zulässige bilineare Abbildung $\hat{e} : G_1 \times G_1 \to G_2$. Wähle einen Erzeuger $P \in G_1$.

- Erzeuge einen (N, c, l, ϵ)-collusion-secure Code $C = \{\omega_1, \ldots, \omega_N\}$.
- Für $j = 1, \ldots, l$ wähle $a, z_j \in \mathbb{Z}_q^\times$ zufällig. Setze $Q = aP, Z_j = g^{z_j}$.
- Wähle eine Hashfunktion $H : G_1 \to M$.

Die Systemparameter sind $(q, G_1, G_2, \hat{e}, P, H)$

Private key: $(a, (z_j)_{j=1,\ldots,l})$

Encryption key: $(g, Q, \{Z_j\}_{j=1,\ldots,l})$

User key: Dem Benutzer u_i wird das Codeword ω_i und $(\alpha_{\omega_{i,j},j}, \beta_{\omega_{i,j},j})$ mit $\alpha_{\omega_{i,j},j} + a\beta_{\omega_{i,j},j}) = z_j mod q$ zugeordnet.

Proxy Quantity: Der Benutzer u_i erhält die Proxy Quantity $\Pi_i = (\Pi_{\omega_{i,1},1}, \ldots, \Pi_{\omega_{i,l},l})$, wobei $\Pi_{\omega_{i,j},j} := (\alpha_{\omega_{i,j},j}, \beta_{\omega_{i,j},j} P)$.

Encrypt: Nachricht $(m_1, \ldots, m_l) \in M^l$. Wähle zufälliges $k \in \mathbb{Z}_q$ und berechne $c_1 = kP, c_2 = k^2P, d_j = m_j \oplus H(Z_j^{k^2})$. Ausgabe $(c_1, c_2, d_1, \ldots, d_l) \in G_1^2 \times G_2^l$.

Decrypt: Aus $(c_1, c_2, d_1, \ldots, d_l) \in G_1^2 \times G_2^l$ berechne $Z_j^{k^2} = \hat{e}(\alpha_{\omega_{i,j},j} c_1, c_1) \times \hat{e}(\beta_{\omega_{i,j},j} P, c_2)$ und $m_j = d_j \oplus H(Z_j^{k^2})$.

Diskussion der Einsatzmöglichkeiten

Warum setzen Pay-TV-Anbieter diese Verfahren bislang nicht ein?

- Traitor-tracing-Verfahren erzeugen einen gewissen Overhead [2], der die Übertragungskosten steigert. Eine Umstellung bestehender Systeme ist ebenfalls mit Kosten verbunden.

- Kosten für einen Satellitentransponder ca. $2,5 \cdot 10^6$ EUR pro Jahr [3]. Die Einnahmen von einem Komplett-Abonnenten eines großen deutschen Pay-TV-Senders betragen ca. $5,4 \cdot 10^2$ EUR pro Jahr [4].

- Verluste durch Verbreitung von geheimen Schlüsseln sind mutmaßlich klein ge-

genüber den Verlusten durch den Einsatz unsicherer Kryptographie (BetaCrypt-Crack führte angeblich zu 500.000 Schwarzsehern [5])

- In Zukunft ist ein Einsatz jedoch denkbar, zumal ein Austausch von Verschlüsselungsverfahren ohnehin gelegentlich notwendig ist (BetaCrypt 2003 [6], SECA-2 2004 [7], evtl. Nagravision)

- Die Situation ändert sich für Internet-Broadcasts, da dort die Menge der zu übertragenden Daten von geringerer Bedeutung ist.

Literatur

[1] H. Chabanne, D. H. Phan, D. Pointcheval: *Public Traceability in Traitor Tracing Schemes*. In Advances in Cryptology - Proceedings of EUROCRYPT '05. Springer-Verlag, 2005.

[2] B. Chor, A. Fiat, M. Naor: *Tracing Traitor*. In Advances in Cryptology - Proceedings of Crypto 1994. Springer, Verlag 1994.

[3] A. Voigt: *Übertragung von breitbandigen Multimediadiensten via Satellit*. Diplomarbeit 1999, Fachhochschule Darmstadt.

[4] Premiere: Abos & Preise im Überblick.
http://www.premiere.de/premweb/cms/de/paketeundpreise_abonnieren.jsp

[5] Premiere wechselt Verschlüsselungssystem. Heise-Verlag, 2003.
http://www.heise.de/newsticker/meldung/35630

[6] Premiere Chronik 2003.
http://info.premiere.de/inhalt/de/unternehmen_chronik_2003.jsp.

[7] NDS Press Room, *Sky Italia to Stop Using SECA Encryption by End 2004*
http://nds.com/press_room/article_sky_italia_290404.html