Hierarchical identity based cryptography, Certificateless public key cryptography

Hannes Mehnert

07.12.2005

Übersicht

- Hierarchical identity based cryptography
- Hierarchical identity based encryption with constant size ciphertext
- Certificateless public key cryptography

Hierarchical identity based cryptography

- Motivation: Dezentralisierung des Private Key Generators (PKG)
- Aufgaben des PKG:
  - Generierung des privaten Schlüssels
  - Verifikation von Identifikationsnachweisen
  - Aufbau verschlüsselter Verbindungen zur Übertragung der privaten Schlüssel
- Idee: Hierarchie von PKG
  - Lastverteilung
  - Vertrauensverteilung
  - Schaden eines kompromittierten PKG lokal begrenzt

BasicHIDE - Root Setup

- IG: BDH parameter Generator
- Input: K
- Generiere mit IG $\mathbb{G}_1$ und $\mathbb{G}_2$ mit primer Ordnung q
- admissible pairing e: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
- Wähle Generator $P_0 \in \mathbb{G}_1$
- Zufällig $s_0 \in \mathbb{Z}/q\mathbb{Z}$, $Q_0 = s_0 P_0$
- H_1: $\{0,1\}^* \rightarrow \mathbb{G}_1$, H_2: $\mathbb{G}_2 \rightarrow \{0,1\}^n$
- $M = \{0,1\}^n$, $C = \mathbb{G}_1^t \times \{0,1\}^n$
- Output: params $= (\mathbb{G}_1, \mathbb{G}_2, \text{e}, P_0, Q_0, H_1, H_2)$
- root PKG secret s_0

BasicHIDE - Lower-level Setup

- Input: params
- Entity E_t wählt ihr secret $s_t \in \mathbb{Z}/q\mathbb{Z}$ zufällig

BasicHIDE - Extraction

- E_t mit ID-Tupel $(ID_1, ..., ID_t)$. S_0 Identität in $\mathbb{G}_1$. E_t parent:
- $P_t = H_1(ID_1, ..., ID_t) \in \mathbb{G}_1$
- $S_t = S_{t-1} + s_{t-1}P_t = \sum_{i=1}^{t} s_{i-1}P_i$
- E_t bekommt $Q_i = s_i P_0$ für $1 \leq i \leq t-1$

BasicHIDE - Encryption

- Input: M, $(ID_1, ..., ID_t)$
- $P_i = H_1(ID_1, ..., ID_i) \in \mathbb{G}_1$ für $1 \leq i \leq t$
- Zufällig $r \in \mathbb{Z}/q\mathbb{Z}$
- $C = [rP_0, rP_2, ..., rP_t, M \oplus H_2(g^r)]$, wobei $g = e(Q_0, P_1) \in \mathbb{G}_2$
- Output: C

BasicHIDE - Decryption

- Input $C = [U_0, U_2, ..., U_t, V]$
- $V \oplus H_2(\frac{e(U_0, S_t)}{\prod_{i=2}^{t} e(Q_{i-1}, U_i)}) = M$

Security - FullHIDE

- Basierend auf BDH
- PKG muß nicht immer dasgleiche s_t für jeden private key benutzen
- H_1 kann eine iterated Hash Funktion sein
- Nicht sicher gegen adaptive chosen ciphertext
- Fujisaki-Okamoto Padding erweitert IBE zu einem IND-CCA sicherem IBE
- Dazu 2 weitere Hashfunktionen im Setup
- $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}/q\mathbb{Z}$
- $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$

FullHIDE - Encryption

- Input: M, $(ID_1, ..., ID_t)$
- $P_i = H_1(ID_1, ..., ID_i) \in \mathbb{G}_1$ für $1 \leq i \leq t$
- zufällig $\sigma \in \{0,1\}^n$
- $r = H_3(\sigma, M)$
- $C = [rP_0, rP_2, ..., rP_t, \sigma \oplus H_2(g^r), M \oplus H_4(\sigma)]$
- wobei $g = e(Q_0, P_1) \in \mathbb{G}_2$

FullHIDE - Decryption

- Input: $C = [U_0, U_2, ..., U_t, V, W]$
- $V \oplus H_2(\frac{e(U_0, S_t)}{\prod_{i=2}^{t} e(Q_{i-1}, U_i)}) = \sigma$
- $W \oplus H_4(\sigma) = M$
- $r = H_3(\sigma, M)$
- Test BasicHIDE verschlüsselt (mit r, C)
- Output: M

Dual-HIDE

- 2000 von Sakai, Ohgishi und Kasahara vorgestellt
- Key sharing scheme
- PKG vergibt private Schlüssel sP_y, s master secret, $P_y = H_1(ID_y)$
- x und y haben durch $e(sP_y, P_z) = e(P_y, P_z)^s = e(P_y, sP_z)$ shared secret

Dual-HIDE

- HIDE kann somit effizienter werden für in der Hierarchie nahe liegende Nutzer
- Nutzer y: $(ID_{y1}, ..., ID_{yl}, ..., ID_{ym})$
- Nutzer z: $(ID_{z1}, ..., ID_{zl}, ..., ID_{zn})$
- $(ID_{y1}, ..., ID_{yl}) = (ID_{z1}, ..., ID_{zl})$

Dual-HIDE Encryption

- Input: M, y
- $P_{zi} = H_1(ID_{z1}, ..., ID_{zi}) \in \mathbb{G}_1$ für $l + 1 \leq i \leq n$
- Zufälliges $r \in \mathbb{Z}/q\mathbb{Z}$
- $C = [rP_0, rP_{z(l+1)}, ..., rP_{zn}, M \oplus H_2(g_{yl}^r)]$
- wobei $g_{yl} = \frac{e(P_0, S_y)}{\prod_{i=l+1}^{m} e(Q_{y(i-1)}, P_{yi})} = e(P_0, S_{yl})$

Dual-HIDE Decryption

- Input: $C = [U_0, U_{l+1}, ..., U_n, V]$
- $V \oplus H_2(\frac{e(U_0, S_z)}{\prod_{i=l+1}^{n} e(Q_{z(i-1)}, U_i)}) = M$

Dual-HIDE

- Aufwand: Encrypter $m - l + 1$, Decrypter $n - l + 1$
- Wenn $m < 2l - 1$, Aufwand geringer Aufwand von HIDE
- Broadcast-Verschlüsselung an alle, deren secret key von demgleichen PKG stammt
- Mit Fujisaki-Okamoto Padding IND-CCA secure

Restricting Key Escrow

- Jeder PKG hat die privaten Schlüssel seiner Nutzer
- Dual-HIDE: PKG kann $S_l = S_l + bP_l$ und $Q_{l-1} = Q_{l-1} + bP_0$ mit einem beliebigen $b \in \mathbb{Z}/q\mathbb{Z}$
- Nur noch PKG kennt ihren secret key

Constant size ciphertext

- Motivation: HIBE ciphertext Größe und decryption Kosten
- Ciphertext Größe und decryption Kosten sind unabhängig von der Tiefe k
- Ciphertext ist drei Gruppenelemente
- Decryption zwei bilineare map Berechnungen
- Sei $\mathbb{G}$ und $\mathbb{G}_1$ multiplikative bilineare Gruppen primer Ordnung p
- $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$

HIBE-CSC - Setup

- Input: l (maximale Tiefe)
- Zufälliger Generator $g \in \mathbb{G}$
- Zufälliges $\alpha \in \mathbb{Z}_p$
- $g_1 = g^{\alpha}$
- Zufällig $g_2, g_3, h_1, ..., h_l \in \mathbb{G}$
- $g_4 = g_2^{\alpha}$
- params $= (g, g_1, g_2, g_3, h_1, ..., h_l)$
- master key $= g_4 = g_2^{\alpha}$

HIBE-CSC - Key generation

- Input: $d_{ID|k-1}$, ID
- private key $d_{ID} = (g_2^{\alpha}(h_1^{I_1}...h_k^{I_k}g_3)^r, g^r, h_{k+1}^r, ..., h_l^r) \in \mathbb{G}^{2+l-k}$

HIBE-CSC - Encrypt

- Input: params, ID, M
- ID $= (I_1, ..., I_k) \in \mathbb{Z}_p^k$
- $CT = (e(g_1, g_2)^s M, g^s, (h_1^{I_1} ... h_k^{I_k} g_3)^s) \in \mathbb{G} \times \mathbb{G}^2$

HIBE-CSC - Decrypt

- Input: d_{ID}, CT
- $CT = (A, B, C)$
- $d_{ID} = (I_1, ..., I_k)$
- $\frac{Ae(a_1, C)}{e(B, a_0)} = M$

HIBE-CSC - Security

- Basierend auf BDHE Problem
- IND-sID-CPA (selective identity secure)
- IND-sID-CCA und IND-ID-CCA kann erreicht werden

Certificateless public key cryptography

- Motivation: Zertifikatslose public-key Cryptosysteme ohne key escrow
- Idee: PKG generiert nur noch einen Teil des privaten Keys
- Dadurch public key und ID

Basic CL-PKE - Setup

- Input k
- Generiere $(\mathbb{G}_1, \mathbb{G}_2, e)$, $\mathbb{G}_1$ und $\mathbb{G}_2$ Gruppen primer Ordnung q, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
- Wähle Generator $P \in \mathbb{G}_1$
- Master-Key s zufällig aus $\mathbb{Z}_q^*$, setze $P_0 = sP$
- Wähle Hashfunktionen $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ und $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$
- Output $params = (\mathbb{G}_1, \mathbb{G}_2, e, n, P, P_0, H_1, H_2)$
- Message space $\mathrm{M} = \{0,1\}^n$
- Ciphertext space $\mathrm{C} = \mathbb{G}_1 \times \{0,1\}^n$

Basic CL-PKE - Partial private key extract

- Input: $ID_A \in \{0,1\}^*$
- $Q_A = H_1(ID_A) \in \mathbb{G}_1^*$
- Output $D_A = sQ_A \in \mathbb{G}_1^*$
- Verifizierbar durch $e(D_A, P) = e(Q_A, P_0)$

Basic CL-PKE - Set secret value

- Input: params, ID_A
- Zufällig $x_A \in \mathbb{Z}_q^*$
- x_A A's secret key

Basic CL-PKE - Set private key

- Input: params, D_A, x_A
- $S_A = x_A D_A = x_a s Q_A \in \mathbb{G}_1^*$

Basic CL-PKE - Set public key

- Input: params, x_A
- $P_A = (X_A, Y_A) = (x_A P, x_A P_0) = (x_A P, x_A s P)$

Basic CL-PKE - Encrypt

- Input: M, P_A
- Teste: $X_A, Y_A \in \mathbb{G}_1^*$ und $e(X_A, P_0) = e(Y_A, P)$
- $Q_A = H_1(ID_A) \in \mathbb{G}_1^*$
- Wähle $r \in \mathbb{Z}_q^*$ zufällig
- $C = (rP, M \oplus H_2(e(Q_A, Y_A)^r))$

Basic CL-PKE - Decrypt

- Input: C = (U, V), S_A
- $V \oplus H_2(e(S_A, u))$

Security CL-PKE

- Angreifer: A_1
  - Hat nicht master-key
  - Kann beliebig public keys abfragen und ersetzen
  - Kann private und partial private keys abfragen
  - Kann decryption queries machen
- Angreifer: A_2
  - Hat master-key
  - Darf keine public keys ersetzen 'item Kann beliebig public keys abfragen
  - Kann partial private keys abfragen
  - Kann decryption queries machen
- Fujisaki-Okamoto Padding erweitert CL-PKE zu einem chosen ciphertext secure CL-PKE
- Full CL-PKE Scheme ist IND-CCA sicher, auf GBDHP rückführbar

CL-PKC Binding

- CL-PKE PKG kann public key verfälschen
- Binding läßt nur einen public key pro Identität zu
- Zuerst secret value x_A und public key P_A festlegen
- Dann wird $Q_A = H_1(ID_A|P_A)$

Hierarchical CL-PKE

- Motivation: Lastverteilung, bisherige hierarchische IDE haben Key Escrow
- Idee: Anwendung von CL-PKE auf HIDE

BasicHCL-PKE - Setup

- Bis auf den Ciphertext space wie BasicCL-PKE:
- Input k
- Generiere $(\mathbb{G}_1, \mathbb{G}_2, e)$, $\mathbb{G}_1$ und $\mathbb{G}_2$ Gruppen primer Ordnung q, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
- Wähle Generator $P \in \mathbb{G}_1$
- Master-Key x_0 zufällig aus $\mathbb{Z}_q^*$, setze $P_0 = x_0 P$
- Wähle Hashfunktionen $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ und $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$
- Output $params = (\mathbb{G}_1, \mathbb{G}_2, e, n, P, P_0, H_1, H_2)$
- Message space $\mathrm{M} = \{0,1\}^n$
- Ciphertext space $C_t = \mathbb{G}_1^t \times \{0,1\}^n$

BasicHCL-PKE - Partial private key extract

- Wird auf Ebene t-1 ausgeführt für Ebene t
- $Q_t = H_1(ID_1|ID_2|...|ID_t) \in \mathbb{G}_1^*$
- $D_t = D_{t-1} + x_{t-1}Q_t = \sum_{i=1}^{t} x_{i-1}Q_i$

BasicHCL-PKE - Set secret value

- Input: params, t, ID-Tupel (ID_1, ID_2, ..., ID_t)
- Wählt zufällig $x_t \in \mathbb{Z}_q^*$
- Output: x_t

BasicHCL-PKE - Set private key

- Gleich wie bei BasicCL-PKE
- private key $S_t = x_t D_t$

BasicHCL-PKE - Set public key

- Gleich wie bei BasicCL-PKE
- public key $P_t = (X_t, Y_t)$
- $Y_t = x_0 X_t = x_0 x_t P$

BasicHCL-PKE - Encryption

- Input: M, t, ID-Tupel $(ID_1, ID_2, ..., ID_t)$
- Teste $e(X_i, P_0) = e(Y_i, P)$ für $1 \leq i \leq t$
- $Q_i = H_1(ID_1|ID_2|...|ID_i) \in \mathbb{G}_1^*$ für $2 \leq i \leq t$
- Zufällig $r \in \mathbb{Z}_q^*$
- $C = (U_0, U_2, ..., U_t, V) =$
  $(rP_0, rQ_2, rQ_3, ..., rQ_t, M \oplus H_2(e(Q_1, Y_t)^r)) \in C_t$

BasicHCL-PKE - Decryption

- Input: $C = (U_0, U_2, ..., U_t, V)$, t, ID-Tupel $(ID_1, ID_2, ..., ID_t)$
- $V \oplus H_2(\frac{e(S_t, U_0)}{\prod_{i=2}^{t} e(x_t X_{i-1}, U_i)})$

Quellen

- Hierarchical ID-Based Cryptography (Craig Gentry, Alice Silverberg)
- Hierarchical Identity Based Encryption with Constant Size Ciphertext (Dan Boneh, Xaier Boyen, Eu-Jin Goh)
- Certificateless Public Key Cryptography (Sattam S. Al-Riyami, Kenneth G. Paterson)