Potsdam, 22.7.2009

Kryptographie und Datensicherheit

Multicast Security & Copyright Protection

Ein Vortrag von Linda Tschepe

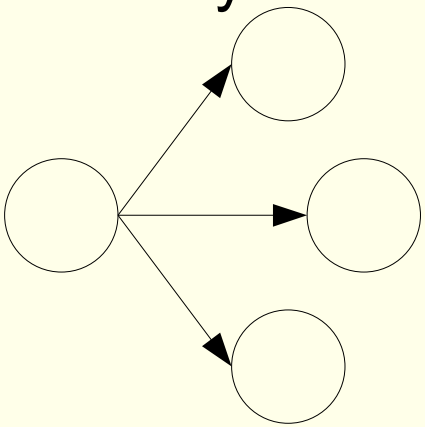
Übersicht

- **Allgemeines**
- **Secret Sharing Schemes**
  - **Shamir Schwellenschema**
- **Broadcast-Verschlüsselung**
- **Multicast Re-keying**
  - **Naor-Pinkas-Re-keying Scheme**
  - **Logical key Hierarchy**
- **Urheberschutz**
  - **Fingerprinting**

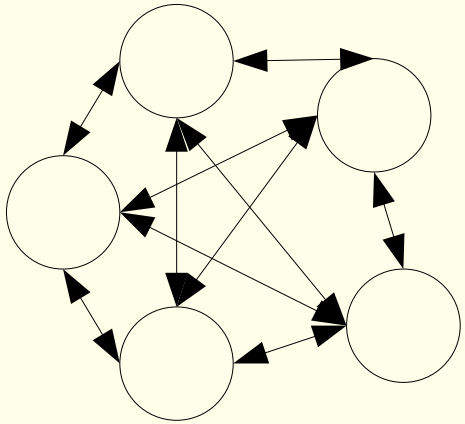
Allgemeines

Multicast bezieht sich auf eine Nachricht, die mehrere Empfänger hat:

One-to-Many-Kommunikation



Many-to-Many-Kommunikation



z.B.: Online Chat Group, Streaming (video & audio), Videokonferenz, Multiplayer Games

Allgemeines

Beim Aufbau von Multicast-Systemen bedürfen folgende Fragen besonderer Beachtung:

- Gibt es einen oder mehrere Sender?
- Ist die Gruppe langlebig oder kurzlebig?
- Ist die Gruppe dynamisch oder statisch?

Secret Sharing

Ziel von Secret Sharing Schemes:
Zerlegen eines Geheimnisses in Teilgeheimnisse, so dass das Geheimnis nur aus bestimmten, vorher festgelegten Gruppen von Teilgeheimnissen rekonstruiert werden kann

Möglicher Einsatz:

- Zugangskontrolle: Das Ergebnis der Rekonstruktion wird mit einem im System gespeicherten Geheimnis verglichen
- Geheimniserzeugung: das rekonstruierte Geheimnis, welches vorher nicht vorlag, wird kryptographisch weiterverarbeitet (z.B. als Signaturschlüssel)

Secret Sharing

Beispiele für Sectret Sharing Schemes:

- (r,v)-Schwellenschemata (Threshold Schemes):
  - Erzeugen von v Teilgeheimnissen aus dem Geheimnis S
  - S kann nur durch r oder mehr Teilgeheimnisse rekonstruiert werden
- Komplexe Zugriffsstrukturen:
  - S kann nur dann rekonstruiert werden, wenn die Teilgeheimnisse aus einer zuvor festgelegten Zugriffsstruktur stammen
  - z.B. zum öffnen des Firmentresors werden entweder 2 Direktoren, 3 Mitarbeiter oder ein Direktor und 2 Mitarbeiter benötigt

Secret Sharing – Shamir Schwellenschema

Das (r,v)-Schwellenschema von Shamir nutzt folgendes:

- K – ein endlicher Körper
- a – ein Polynom vom Grad r-1 in K[x]
- Wenn man mindestens r Punkte von a kennt, kann man a eindeutig rekonstruieren

Secret Sharing – Shamir Schwellenschema

Beispiel:

$K=Z_{17}$, S=4, v=7, r=3 -> $a(x)= a_2 x^2 + a_1 x + a_0$

Das Geheimnis S wird als Y-Achsenabschnitt festgesetzt.

$S = a_0 = 4$

$a_1 = 3$, $a_2 = 1$ **$a(x) = x^2 + 3x + 4$**

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a(x)	4	8	14	5	15	10	7	6	7	10	15	5	14	8	4	2	2

Man wähle nun v dieser Punkte als Teilgeheimnisse aus.

Secret Sharing – Shamir Schwellenschema

Rekonstruktion:

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a(x)	4	8	14	5	15	10	7	6	7	10	15	5	14	8	4	2	2

Lösen des Gleichungssystems:

1: $a_2 + a_1 + a_0 = 8$

2: $a_2\, 6^2 + a_1\, 6 + a_0 = 7$

3: $a_2\, 3^2 + a_1\, 3 + a_0 = 5$

Secret Sharing – Shamir Schwellenschema

1: $a_2 + a_1 + a_0 = 8$

2: $a_2\ 2 + a_1\ 6 + a_0 = 7$

3: $a_2\ 9 + a_1\ 3 + a_0 = 5$

15 **1** + **2**

8 **1** + **3**

1: $a_2 + a_1 + a_0 = 8$

2: $a_1\ 4 + a_0\ 16 = 1$

3: $a_1\ 11 + a_0\ 9 = 1$

2 + 5 **3**

1: $a_2 + a_1 + a_0 = 8$

2: $a_1\ 4 + a_0\ 16 = 1$

3: $a_0\ 10 = 6$

$a_0\ 10 = 6$ | * 12

$a_0 =$ **4**

Ein BES (Broadcast Encryption Scheme)

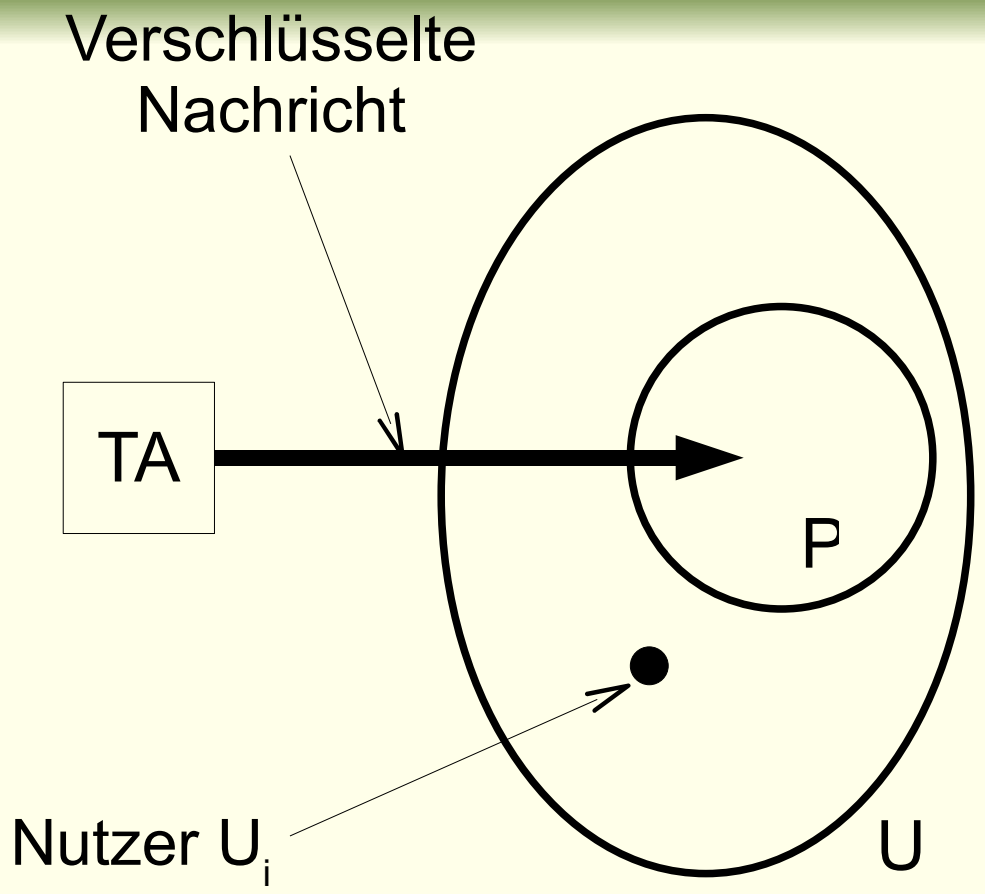
U – gesamtes Netzwerk

n – Anzahl der Nutzer in U

P – privilegierte Gruppe

$P \subseteq U$

TA – vertrauenswürdige Autorität (trusted Authority)



Ein triviales BES - Aufbau

1. Aufbau Phase

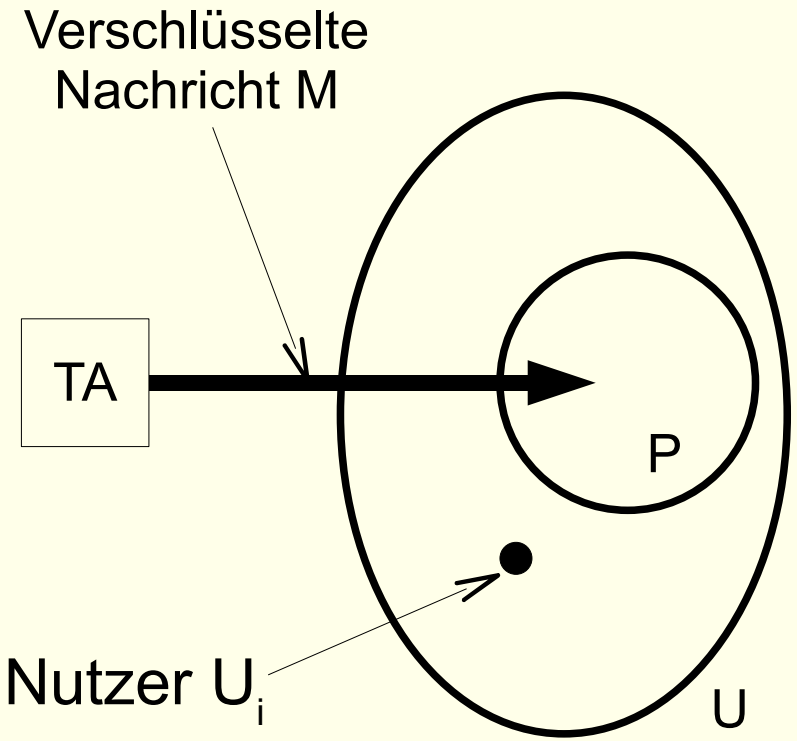
jeder Nutzer U_i erhält einen Schlüssel K_i

2. Verschlüsselung des Schlüssels

K wird verschlüsselt gesendet mit
$y_i = e_{Ki}(K)$ (alle K_i von P)

3. Verschlüsselung der Nachricht

$y = e_K(M)$



Ein triviales BES – Vor- und Nachteile

Vorteile	Nachteile
hohe Sicherheit (kein Unpreviligierter kennt K)	hohe Nachrichten Expansion
geringer Speicherbedarf (pro Nutzer nur ein Schlüssel)	

Ein generelles BES - Aufbau

1. **Aufbauphase**
   Verteilen von Schlüsselmaterial von v Schlüsselvorverteilungsschemata
2. **Secret Sharing**
   der Schlüssel K wird unter Verwendung eines (r,v)-Schwellenschemas in v Teilgeheimnisse geteilt
3. **Verschlüsselung der Teilgeheimnisse**
   jedes s_i wird mit den entsprechenden Gruppenschlüsseln von P verschlüsselt und versandt
4. **Verschlüsselung der Nachricht**
   $y = e_K(M)$

Ein generelles BES – 1. Aufbauphase

Beispiel:
n = |U| = 7, v = 7, r = 3

1. Konstruktion von v Fiat-Naor-1-KDPs $F_1,...,F_v$, wobei jedes F_i auf einer anderen Teilmenge von U definiert ist

F_i ist auf je drei Nutzern U_a, U_b, U_c definiert:

(jede Teilmenge beinhaltet r Nutzer)

	U_a	U_b	U_c
I_i	1	1	1
I_{ia}	0	1	1
I_{ib}	1	0	1
I_{ic}	1	1	0

Ein generelles BES – 1. Aufbauphase

2. eine vxn-Matrix besagt, welches F_i mit welchem Nutzer U_j assoziiert wird

U_j werden die Schlüsselmaterialien von F_i gegeben $\Leftrightarrow$ M[i,j] = 1

$users(F_i) = \{ U_j \mid M[i,j] = 1\}$

$schemes(U_j) = \{ F_i \mid M[i,j] = 1\}$

Für M müssen folgende Eigenschaften gelten:

- Für alle Pärchen (U_j , $U_{j'}$) existiert genau ein KDP mit $\{U_j, U_{j'}\} \subseteq users(F_i)$
- M ist eine vxn-Matrix mit exakt r „1“-en in jeder Spalte

Ein generelles BES – 1. Aufbauphase

$$M = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

	U_a	U_b	U_c
I_i	1	1	1
I_{ia}	0	1	1
I_{ib}	1	0	1
I_{ic}	1	1	0

$users(F_1) = \{ U_1, U_2, U_4 \}$

- U_1 erhält I_1 , $I_{1,2}$, $I_{1,4}$
- U_2 erhält I_1 , $I_{1,1}$, $I_{1,4}$
- U_4 erhält I_1 , $I_{1,1}$, $I_{1,2}$

Ein generelles BES – 1. Aufbauphase

3.: Jeder Nutzer U_j erhält seine Schlüssel

U_1	U_2	U_3	U_4	U_5	U_6	U_7
I_1	I_1	I_2	I_1	I_2	I_3	I_4
$I_{1,2}$	$I_{1,1}$	$I_{2,2}$	$I_{1,1}$	$I_{2,2}$	$I_{3,3}$	$I_{4,4}$
$I_{1,4}$	$I_{1,4}$	$I_{2,5}$	$I_{1,2}$	$I_{2,3}$	$I_{3,4}$	$I_{4,5}$
I_5	I_2	I_3	I_3	I_4	I_5	I_6
$I_{5,5}$	$I_{2,3}$	$I_{3,4}$	$I_{3,3}$	$I_{4,4}$	$I_{5,1}$	$I_{6,2}$
$I_{5,6}$	$I_{2,5}$	$I_{3,6}$	$I_{3,6}$	$I_{4,7}$	$I_{5,5}$	$I_{6,6}$
I_7	I_6	I_7	I_4	I_5	I_6	I_7
$I_{7,3}$	$I_{6,6}$	$I_{7,1}$	$I_{4,5}$	$I_{5,1}$	$I_{6,2}$	$I_{7,1}$
$I_{7,7}$	$I_{6,7}$	$I_{7,7}$	$I_{4,7}$	$I_{5,6}$	$I_{6,7}$	$I_{7,3}$

Ein generelles BES – 2. Secret Sharing

Teilung des Schlüssels K in v Teile $s_1,...,s_v$ mit Hilfe eines (r,v)-Schwellenschemas

Hierzu kann man zum Beispiel das bereits vorgestellte Shamir-Schwellenschema Nutzen.

Ein generelles BES – 3. Verschlüsselung K

Jedes s_i wird mit einem K_i verschlüsselt.

$$K_i = I_i + \sum_{\{j|\ Uj \in users(Fi) \setminus P\}} I_{i,j}$$

$$P = \{U_1, U_5, U_7\}$$

$$M = \begin{pmatrix} 1&1&0&1&0&0&0 \\ 0&1&1&0&1&0&0 \\ 0&0&1&1&0&1&0 \\ 0&0&0&1&1&0&1 \\ 1&0&0&0&1&1&0 \\ 0&1&0&0&0&1&1 \\ 1&0&1&0&0&0&1 \end{pmatrix}$$

K_1	$I_1 + I_{1,2} + I_{1,4}$
K_2	$I_2 + I_{2,2} + I_{2,5}$
K_3	$I_3 + I_{3,3} + I_{3,4} + I_{3,6}$
K_4	$I_4 + I_{4,4} + I_{4,7}$
K_5	$I_5 + I_{5,6}$
K_6	$I_6 + I_{6,6}$
K_7	$I_7 + I_{7,3}$

Ein generelles BES – 4. Verschlüsselung M

Verschlüsselung der Nachricht M mit K und Senden des Broadcasts $b_P + y$

$$y = e_K(M)$$

$$b_P = \{ e_{Ki}(s_i) \mid 1 \leq i \leq 7\}$$

Nachrichtenexpansion ist abhängig von der Anzahl der Shares.

Multicast Re-keying

- langlebige dynamische Gruppe
  - ➔ meist ein Gruppenschlüssel zum Verschlüsseln der Nachricht und für jeden Nutzer long-lived-keys zum Verschlüsseln des Gruppenschlüssels
  - ➔ 2 Operationen notwendig:
    - user join operation:
      - wenn ein Nutzer zur Gruppe stößt muss er seine persönlichen LL-keys erhalten, sowie den aktuellen Gruppenschlüssel
    - user revocation operation:
      - verteilt an die verbleibenden Gruppenmitglieder einen neuen Gruppenschlüssel und evtl. auch neue LL-keys

Re-keying – Naor-Pinkas-Re-keying scheme

- basiert auf dem Shamir Schwellenschema

$\{U_1,..., U_n\} = U$, $|U| = n$,

F – Menge der ungültigen Nutzer, $F \subset U$, $|F| = w$

K' – neuer Gruppenschlüssel

1. Initialisierungsphase:
   - Erzeugung von n Teilgeheimnissen $y_1,..., y_n$ von K' mit Hilfe eines Shamir (w+1,n)-Schwellenschemas.
   - jeder $U_i \in U$ erhält y_i
2. Revocation:
   - versenden aller Teilgeheimnisse y_j mit $U_j \in F$

Re-keying – Naor-Pinkas-Re-keying scheme

- Es ist auch möglich w' < w Nutzer aus der Gruppe zu entfernen:
  - Es werden dann die w' y_i gesendet, sowie w-w' neue Teilgeheimnisse, die noch nicht in Nutzung sind

- Speicherbedarf: *O(1)*
- Größe des Broadcasts: *O(w)*

Es ist also möglich bis zu w Nutzer auszuschließen, dies kann jedoch nicht stufenweise geschehen.
-> Verbesserung (siehe Stinson S. 535)

Re-keying – Logical Key Hierarchy

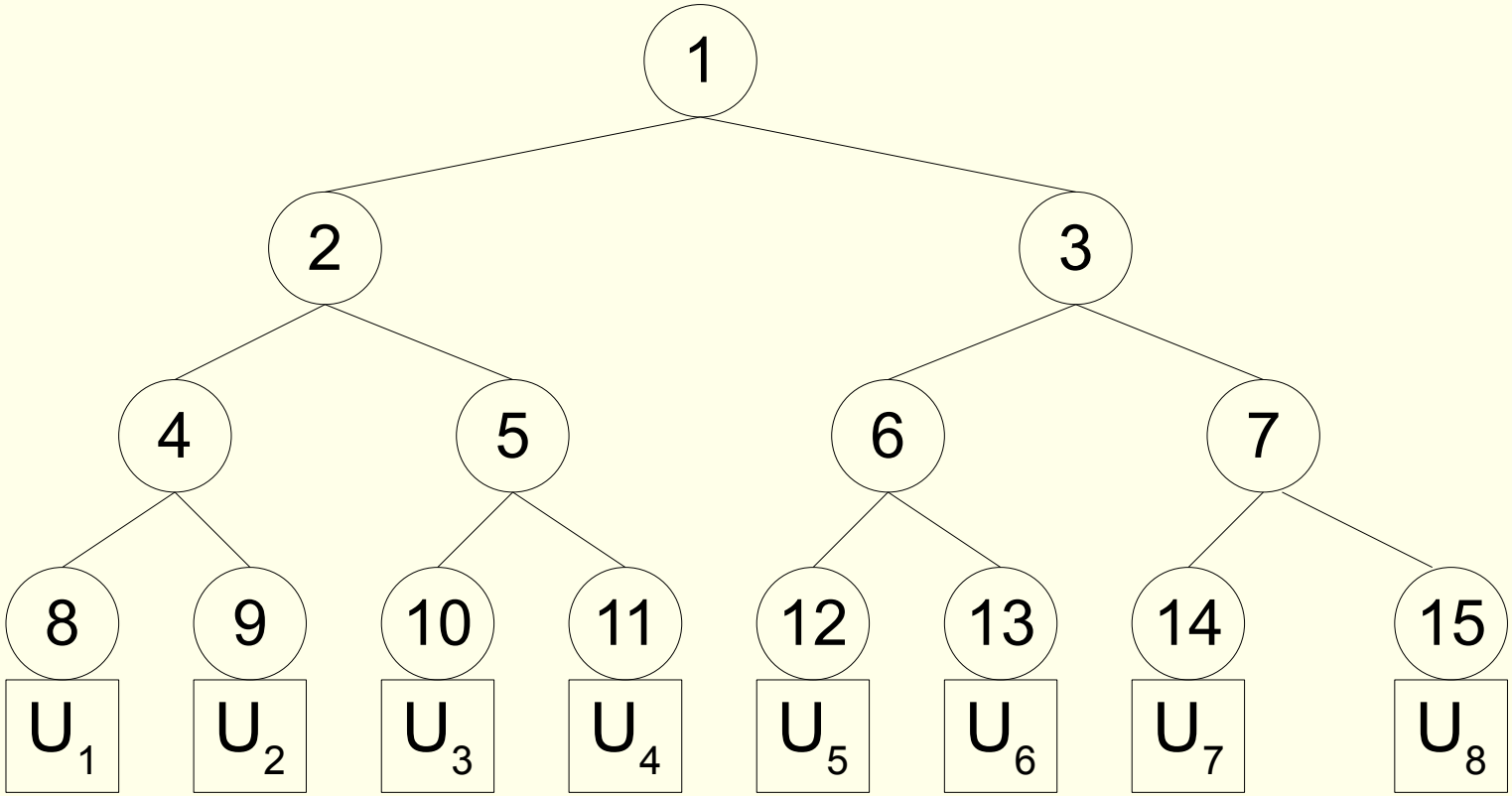
- $|U| = n, \quad 2^{d-1} < n <= 2^d$
- basiert auf einem binärer Baum mit n Blättern und Tiefe d
- jeder Nutzer U_i wird mit einem Blatt assoziiert und umgekehrt
- jeder Knoten X hat einen anderen Schlüssel k(X)

Benennung der Knoten:

1. Knoten auf Level l der Reihe nach: $2^l, 2^l+1, ..., 2^{l+1}-1$
2. Blätter: $2^d, 2^d+1, ..., 2^{d+1}-1$
3. Elternknoten von X (X!=1) ist $\lfloor X/2 \rfloor$
4. linkes Kind von X ist 2X und das rechte 2X+1
5. linker Nachbar von X ist X-1 der rechte X+1

Re-keying – Logical Key Hierarchy

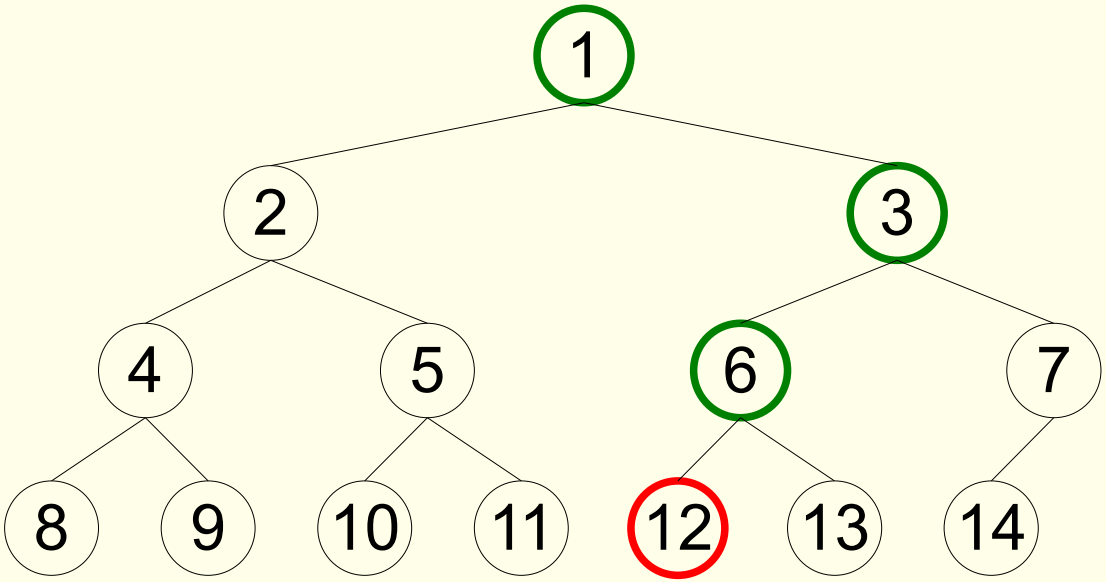
$n = 7 \quad \rightarrow \quad 2^2 < n < 2^3 \quad \rightarrow \quad d = 3$



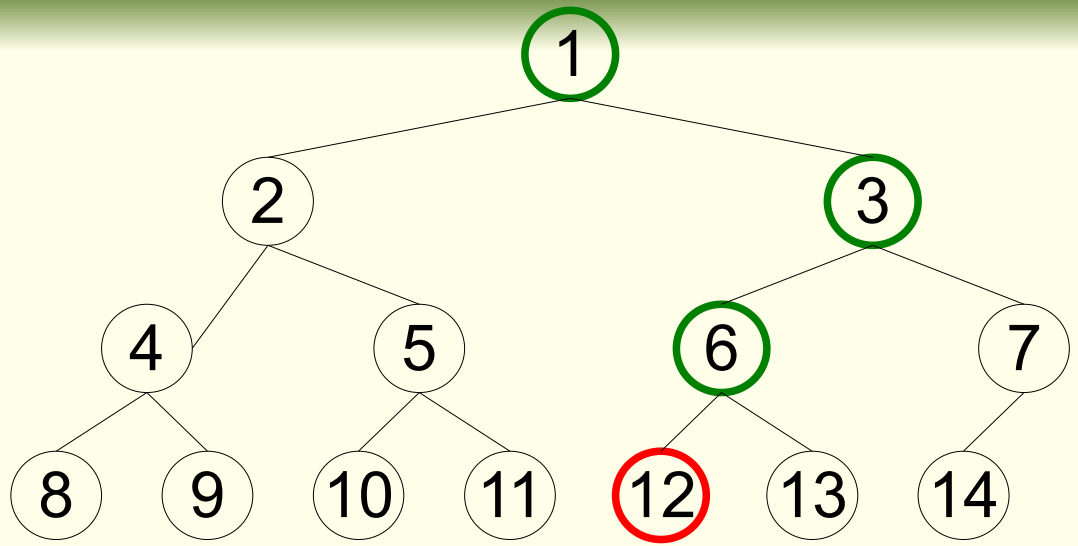
Re-keying – Logical Key Hierarchy

Entfernen eines Nutzers U_i:

Ändere für alle inneren Knoten X und die Wurzel auf dem Pfad zu U_i k(X) in k'(X) um.



Re-keying – Logical Key Hierarchy



|U| = n
d – Tiefe des Baums
$P(U_i)$ – alle Schlüssel von U_i
sib(X) – Zwilling von X
par(X) – Elternknoten von X

Zum Entfernen eines Nutzers U_i assoziiert mit X sendet die TA:

1. $e_{k(sib(X))}(k'(par(X)))$
2. für alle Y in $P(U_i)/\{X, 1\}$:
   $e_{k(sib(Y))}(k'(par(Y)))$ und $e_{k'(Y)}(k'(par(Y)))$

Re-keying – Logical Key Hierarchy

- Speicherbedarf: $O(\log n)$
- Größe des Broadcasts: $O(\log n)$

Diese Größen sind zwar erheblich Größer als die des Naor-Pinkas-Re-keying Schemes, jedoch können beliebig viele Nutzer auf ein mal oder aber auch in Stufen entfernt werden ohne die Sicherheit des Gesamtsystems zu gefährden.

Copyright Protection – Einführung

- heute ist das Kopieren von Inhalt ohne großen Aufwand möglich
  - illegales Versenden des Inhalts nach der Entschlüsselung
  - illegale Weitergabe des Schlüssels, sowie dem Entschlüsselungsalgorithmus
- 2 Lösungsansätze:
  - hardwarebasierte Lösungen (z.B.: tramper resistant devices)
  - Tracing-Algorithmen

Fingerprinting (Watermarking)

- soll gegen die illegale Inhaltsweitergabe wirken
- in digitalen Daten D können zusätzliche Bits enthalten
- zu jedem D_i existiert ein eindeutiger Fingerprints F_i
- der Anbieter speichert alle rechtlichen Eigentümer, sowie deren Fingerprints
- bei einer Piratenkopie könnte der Übeltäter anhand dieses Fingerprints ermittelt werden

Fingerprinting - Angriff

D – Digitale Daten
F – Fingerprint
Cp – Kopie des Inhalts

$D_i = (Cp , F_i)$

- wenn Fingerprintbits in einem D leicht zu entdecken sind, können sie beim Kopieren einfach geändert werden
  - ➔ Bits müssen „versteckt“ werden

- Beim Vergleich zweier D_i werden in jedem Fall Teile des F_i entdeckt
  - ➔ diesen Teil möglichst gering halten bzw.
  - ➔ F_i so konstruieren, dass Tracing trotzdem möglich

Fingerprinting - Angriff

Q – Alphabet
q = |Q|
C - Menge der Codewörter über Q
n = |C|
l – Länge eines Codeworts - $C \subseteq Q^l$
$C_0 \subseteq C$

desc(C_0) – alle hybriden Fingerprints, die aus C_0 erzeugt werden können

$$desc_w(C) = \bigcup_{C0 \subseteq C,\ |C0| <= w} desc(C_0)$$

Fingerprinting - „identifizierbare Eltern“

Wer kann den hybriden Fingerprint erzeugt haben?

Q – Alphabet
q = |Q|
C - Menge der Codewörter über Q
n = |C|
l – Länge eines Codeworts - $C \subseteq Q^l$
$C_0 \subseteq C$

Sei $f \in desc_w(C)$, dann ist:

$$susp_w(f) = \{C_0 \subseteq C : |C_0| <= w, f \in desc(C_0)\}$$

Fingerprinting - „identifizierbare Eltern“

Wenn $|susp_w(f)| = 1$, so sind diese Codes die Erzeuger von f.

Voraussetzung ist allerdings, dass die Koalition nicht größer als w ist.

Unter selbiger Voraussetzung gilt: wenn

$$\bigcap_{C0 \in suspw(f)} C_0 \neq \emptyset$$

dann sind die Elemente dieser Menge identifizierbare Eltern von f.

Fingerprinting - w -IPP Code

wenn gilt:

$$\forall f. \quad \bigcap_{C0 \in suspw(f)} C_0 \neq \emptyset \qquad (f \in desc_w(C))$$

so nennt man C einen w-identifiable parent property code (w-IPP Code).

Fingerprinting - 2-IPP Code ***

Def.: Eine (n,m,w)-perfekte Hash-Familie ist eine Menge F von Funktionen, so dass gilt:

- $|X| = n$.
- $|Y| = m$.
- $\forall f \in F.\ f: X \rightarrow Y$.
- $\forall\ X_1 \subseteq X \mid |X_1| = w.\ \exists\ f \in F\ \forall\ x,x' \in\ X_1 \mid x \neq x'.\ f(x) \neq f(x')$

Wenn $|F| = N$, so schreibt man auch PHF(N;n,m,w).

n x N – Matrix
n = 4
N = 3
w = 2 / 3

1	2	4
3	5	0
0	5	4
1	2	0

Fingerprinting - 2-IPP Code ***

Def.: Eine $(n,m,\{w_1,w_2\})$-separierende Hash-Familie ist eine Menge F von Funktionen, so dass gilt:

- $|X| = n, \quad |Y| = m. \quad \forall f \in F.\ f: X \to Y.$
- $\forall\ X_1, X_2 \subseteq X \mid |X_1| = w_1,\ |X_2| = w_2,\ X_1 \cap X_2 = \emptyset.\ \exists\ f \in F.$
  $(\{f(x) \mid x \in X_1\} \cap \{f(x) \mid x \in X_2\} = \emptyset\ ).$

Wenn $|F| = N$, so schreibt man auch $SHF(N;n,m,\{w_1,w_2\})$.

n x N – Matrix
n = 4
N = 3
w_1/w_2 = 2 / 2

1	2	4
3	5	0
0	5	4
1	2	0

1	2	4
0	3	5
2	0	1
3	2	0

Fingerprinting - 2-IPP Code ***

Man schreibe einen (l,n,q)-Code C als eine n x l Matrix A(C):

Angenommen A(C) ist keine PHF(l;n,q,3), so existieren drei Zeilen r_1, r_2, r_3, welche die PHF-Eigenschaft nicht erfüllen, die also in jeder Spalte c ein Element f_c haben, das in mindestens zwei dieser Zeilen vorkommt.

Es gibt also einen hybriden Fingerprint f, so dass gilt:

$$\{r_1, r_2\}, \{r_2, r_3\}, \{r_1, r_3\} \in \text{susp}_2(f).$$

Somit ist der Schnitt der Mengen aus $\text{susp}_2(f)$ leer und folglich ist C kein 2-IPP Code

Fingerprinting - 2-IPP Code ***

Man schreibe einen (l,n,q)-Code C als eine n x l Matrix A(C):

Angenommen A(C) ist keine SHF(l;n,q,{2,2}), so existieren zwei Mengen aus je zwei Zeilen $\{r_1, r_2\}$, $\{r_3, r_4\}$, welche die SHF-Eigenschaft nicht erfüllen, die also in jeder Spalte c ein Element f_c haben, das sowohl in r_1 oder r_2 als auch in r_3 oder r_4 vorkommt.

Es gibt also einen hybriden Fingerprint f, so dass gilt:

$$\{r_1, r_2\}, \{r_3, r_4\} \in \text{susp}_2(f).$$

Somit ist der Schnitt der Mengen aus $\text{susp}_2(f)$ leer und folglich ist C kein 2-IPP Code

Fingerprinting - 2-IPP Code ***

Ein (l, n, q)-Code C ist genau dann ein 2-IPP-Code, wenn A(C) gleichzeitig eine PHF(l; n, q, 3) und eine SHF(l; n, q, {2,2}) ist.

Identifikation der Eltern: $f \in desc_2(C) \setminus C$

1. $susp_2(f)$ besteht aus einer Menge von 2 Codewörtern
2. $susp_2(f)$ hat in allen Mengen das selbe Codewort

Algorithmus: (Komplexität: $\Theta(n^2)$)

- berechne alle $\binom{n}{2}$ Teilmengen von 2- Codewörtern
- Prüfe für jede Teilmenge {e,f} ob gilt: $f \in desc(\{e,f\})$

Fingerprinting - 2-IPP Code

Konstruktion eines 2-IPP Codes mit Codewortlänge l=3:
(nach Hollmann, van Lint, Linnartz und Tolhuizen)

- wähle $r >= 2$ und $q = r^2 + 2r$
- $S = \{1, ..., r\}$ $(|S| = r)$
- $M = \{r+1, ..., 2r\}$ $(|M| = r)$
- $L = \{2r+1, ..., q\}$ $(|L| = r^2)$

- $C_1 = \{ (s_1, s_2, rs_1 + s_2 + r) \mid s_1, s_2 \in S\} \subseteq S \times S \times L$
- $C_2 = \{ (m, sr + m, s) \mid m \in M, s \in S\} \subseteq M \times L \times S$
- $C_3 = \{ (rm_1 + m_2 - r^2, m_1, m_2) \mid m_1, m_2 \in M\} \subseteq L \times M \times M$

Fingerprinting - 2-IPP Code

Identifikation der Eltern von $f = (f_1, f_2, f_3)$:

1. $f_1 \vee f_2 \vee f_3 \in L$
   - $f_2 \in L \quad \rightarrow \quad r*s + m = f_2, \quad s \in S, m \in M$
   - $f_1 \in L \quad \rightarrow \quad r*m_1 + m_2 + r^2 = f_1, \quad m_1, m_2 \in M$
   - $f_3 \in L \quad \rightarrow \quad r*s_1 + s_2 + r = f_3, \quad s_1, s_2 \in S$
2. $f_1 \wedge f_2 \wedge f_3 \notin L$
   - berechne $i \neq j$, so dass Eltern aus C_i und C_j sind
   - das Elternteil, das zwei Koordinaten gegeben hat kann nun eindeutig identifiziert werden

Fingerprinting - 2-IPP Code

Für jedes r >= 2 existiert ein $(3, 3r^2, r^2+2)$-Code,
der ein 2-IPP Code ist und dessen
Parent Identification Algorithm die Komplexität $O(1)$ hat.

Referenzen

Douglas R. Stinson: Cryptographie: Theory and Practice
3rd Edition, Chapman& Hall\CRC, 2006

Albrecht Beutelspacher, Heike B. Neumann, Thomas Schwarzpaul: Kryptografie in Theorie und Praxis: mathematische Grundlagen für elektronisches Geld, Internetsicherheit und Mobilfunk Vieweg+Teubner Verlag, 2005

Albrecht Beutelspacher, Jörg Schwenk, Klaus-Dieter Wolfenstetter: Moderne Verfahren der Kryptographie: von RSA zu Zero-Knowledge
6..Auflage, Vieweg Verlag, 2006


Ende