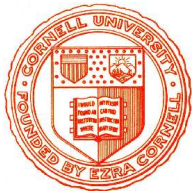
Kryptographie und Komplexität



Einheit 6.4

Mehrparteien-Berechnung und Verteilte Geheimnisse

1. Secret-Sharing Protokolle
2. Verifizierbare Geheimnisaufteilung
3. Threshold Signaturverfahren

Weitere Anwendungen der Kryptographie

Mehrparteien-Berechnungen

- **Verteilung von Information auf eine Gruppe**
  - Aktion wird von mehreren Teilnehmer gemeinsam durchgeführt
    - Öffnen eines Schließfachs in einer Bank
    - Verpacken von Bargeld in der Bundesdruckerei
    - Auszählung von Wahlergebnissen / Elektronische Wahlen
    - Kritische militärische Aktionen
  - Kein einzelner Teilnehmer kann Aktion alleine ausführen
- **Secret Sharing**
  - Absicherung eines Geheimnisses gegen Verrat durch einzelne
    - Einzelpersonen haben nur Teilinformationen
    - Geheimnis kann aus Teilinformation rekonstruiert werden
  - Schutz der Gruppe gegen Blockade durch einzelne
    - Geheimnis kann von Teilgruppe rekonstruiert werden
  - Sicherheitsschwelle abhängig von Bedeutung des Geheimnisses
- **Sichere Funktionenauswertung**
  - Auswertung einer Funktion mit Daten aller Gruppenteilnehmer
  - Ein-/Ausgaben einzelner Teilnehmer bleiben vor anderen verborgen

Secret-Sharing Verfahren

Sicherheit in einer korrumpierbaren Welt

- **Aufteilung eines Geheimnisses**
  - Geheime Information wird auf n Personen verteilt
  - Geheimnis kann nur durch Zusammenarbeit rekonstruiert werden
  - Geheimnis bleibt sicher, auch wenn einzelne unehrlich sind
- **Schwellenschema (Threshold Schemes)**
  - Geheimnis kann nur wiederhergestellt werden, wenn mindestens t Personen beteiligt sind
  - Handlungsfähigkeit bleibt erhalten auch einzelne nicht mitmachen
- **Komplexe Zugriffsstrukturen**
  - Berücksichtigt unterschiedliche Rollen einzelner Personen
  - Geheimnis kann wiederhergestellt werden, wenn eine bestimmte Mindestkonstellation von Personen beteiligt ist
    - z.B. Zwei Abteilungsleiter oder
      Ein Abteilungsleiter und zwei Referenten aus anderen Gruppen

Das Shamir Secret-Sharing Protokoll

- **Verwendet Konstruktion von Polynomsplines**

  Satz: *Ein Polynom $f \in K[x]$ vom Grad $t-1$ ist eindeutig durch t Punkte $y_i = f(x_i)$ bestimmt*

  – Die Lagrange-Interpolationsformel für Polynome liefert die Gleichung
  $$f(x) = \sum_{i=1}^{t} y_i \cdot \prod_{j \neq i} (x - x_i)/(x_i - x_j)$$
  – Bei weniger als t Interpolationspunkten ist eines der y_i unbestimmt und die Gleichung für f hat viele Lösungen

- **Initialisierung für n Personen**

  – Wähle Primzahl $p > n$ und veröffentliche n verschiedene Zahlen $x_i \in \mathbb{Z}_p^*$

- **Verteilung eines Geheimnisses $s \in \mathbb{Z}_p$**

  – Wähle geheime Koeffizienten $a_j \in \mathbb{Z}_p$ und setze $f(x) = s + \sum_{j=1}^{t-1} a_j x^j$

  – Vergebe an den i-ten Geheimnisträger den Geheimnisteil $y_i = f(x_i)$

- **Rekonstruktion des Geheimnisses**

  – Bei t Teilnehmern kann f eindeutig wiederhergestellt werden

  – Das Geheimnis s ist der Wert von f an der Stelle 0

  – Bei weniger als t Teilnehmern gibt es p mögliche Lösungen für s

Wenn Dealer / Teilnehmer sich falsch verhalten

- **Mögliches Fehlverhalten des Dealers**
  - Falsche Berechnung der Teilgeheimnisse y_i
  - Offenlegung der Teilgeheimnisse y_i
  - Ausgabe von weniger als t Teilgeheimnissen
- **Mögliches Fehlverhalten der Teilnehmer**
  - Weigerung, ihr Teilgeheimnis preiszugeben
  - Angabe eines falschen Wertes für das Teilgeheimnis
- **Einfaches Shamir Protokoll bietet keinen Schutz**
  - Fehlverhalten führt zu Geheimnisverrat oder Nichtrekonstruierbarkeit
  - Secret-Sharing Protokolle müssen robust sein
    - Gegen kleine Menge betrügerischer / nichtkooperativer Teilnehmer
    - Gegen betrügerische Dealer

Verifizierbare Geheimnisaufteilung

Identifiziere Fehlverhalten der Beteiligten

- **Dealer muß Geheimnis verschlüsselt preisgeben**
  - Geheimnis und Fragmente werden mit Einwegfunktion chiffriert
  - Für das (t, n) Secret-Sharing Protokoll wählt Dealer geheime Koeffizienten $a_j \in \mathbb{Z}_p$, setzt $f(x) = s + \sum_{j=1}^{t-1} a_j x^j$ und veröffentlicht Werte $u_j = g^{a_j}$, x_i und $z_i = g^{f(x_i)}$ für ein erzeugendes g
  - Der i-te Geheimnisträger erhält Geheimnisteil $y_i = f(x_i)$
- **Kontrolle des Dealers durch die Geheimnisträger**
  - Jeder kann prüfen, ob $g^{\sum_{j=0}^{t-1} a_j (x_i)^j} = \prod_{j=0}^{t-1} u_j^{(x_i)^j} = z_i = g^{f(x_i)}$ ist, d.h. ob die veröffentlichten Punkte von f zu den Koeffizienten passen
  - Teilnehmer i kann prüfen, ob $g^{y_i} = z_i$ gilt, d.h. ob sein Teilgeheimnis wirklich $f(x_i)$ ist
- **Kontrolle der Geheimnisträger durch andere**
  - Bei Rekonstruktion des Geheimnisses kann jeder feststellen, ob Teilnehmer i sein Teilgeheimnis korrekt preisgegeben hat

Threshold Signaturverfahren

Secret-Sharing ohne Preisgabe von Teilgeheimnissen

- **Teilgeheimnisse sind nicht wiederverwendbar**
  - Teilnehmer müssen ihr Teilgeheimnis für die Rekonstruktion des Geheimnisses preisgeben
  - Teilnehmerdaten liegen ab dann für alle anderen offen
  - Für nächste Anwendung müsste neues Geheimnis verteilt werden
  - Ungeeignet für viele Anwendungen verteilter Geheimnisse
- **Signatur mit (t, n) Schwellenschemata**
  - Teilnehmer erzeugen Teilunterschrift, ohne ihre Daten preiszugeben
  - Teile werden durch Combiner oder via Broadcast zusammengesetzt

Threshold Signaturschema mit ElGamal

- **Initialisierung für n Personen**
  - Wähle eine kollisionsresistente Hashfunktion h
  - Wähle große Primzahl $p > n$ und ein erzeugendes Element g von $\mathbb{Z}_p^*$
  - Wähle eine weitere Primzahl q, die $p-1$ teilt
  - Veröffentliche n verschiedene Zahlen $x_i \in \mathbb{Z}_q^*$
- **Verteilung eines Geheimnisses $s \in \mathbb{Z}_q$**
  - Wähle geheime Koeffizienten $a_j \in \mathbb{Z}_q$ und setze $f(x) = s + \sum_{j=1}^{t-1} a_j x^j$
  - Veröffentliche $y = g^s \bmod p$ als Verifikationsschlüssel der Signatur
  - Wähle zufällige $u_i \in \mathbb{Z}_q$ und berechne die Geheimnisteile $s_i = u_i + f(x_i)$
  - Veröffentliche $y_i = g^{s_i} \bmod p$ und $z_i = g^{u_i} \bmod p$
    als Verifikationsschlüssel für Teilnehmer i
- **Verteilung der Informationen**
  - Öffentlich: Hashfunktion h, Primzahlen p, q Verifikationsschlüssel y
    Element g, Stützpunkte x_i, Teilnehmerschlüssel y_i und z_i
  - Teilnehmer i: Geheimnisanteil s_i
  - Dealer: Geheimnis s, Koeffizienten a_i, Zufallswerte u_i

Threshold Signaturschema mit ElGamal (II)

- **Signieren einer Nachricht x durch t Teilnehmer**
  - Teilnehmer i wählt ein zufälliges $k_i \in \mathbb{Z}_q$ und berechnet $r_i = g^{k_i} \bmod p$
  - Jeder erhält die r_i und berechnet $r = \prod_{i=1}^{t} r_i \equiv g^{\sum_{i=1}^{t} k_i} \bmod p$
    sowie $e = h(x, r) \bmod q$ und $\sigma_i = s_i \cdot \prod_{j \neq i} x_j \cdot (x_j - x_i)^{-1} + k_i \cdot e \bmod q$
- **Combiner setzt Teilsignaturen zusammen**
  - Prüfe Signatureteile: $g^{\sigma_i} \equiv y_i^{\prod_{j \neq i} x_j \cdot (x_j - x_i)^{-1}} \cdot r_i^e \bmod p$
  - Gesamtsignatur der Nachricht ist $\sigma = \prod_{i=1}^{t} \sigma_i \bmod q$
- **Verifikation einer Signatur**
  - Berechne $t = \prod_{i=1}^{t} z_i^{\prod_{j \neq i} x_j \cdot (x_j - x_i)^{-1}} \bmod p$ sowie $e = h(x, r) \bmod q$
  - Prüfe ob die Kongruenz $g^{\sigma} \equiv y \cdot t \cdot r^e \bmod p$ gilt

Weitere Fragestellungen der Kryptographie

- **Schlüsselverwaltung**
  - Kryptographische / organisatorische Maßnahmen für sichere Schlüssel
  - Erzeugung, Verteilung und sichere Speicherung von Schlüssel
  - Rückruf kompromittierter Schlüssel
  - Public-Key Infrastrukturen und Zertifizierungsinstanzen
- **Schlüsseletablierungsprotokolle**
  - Protokolle für Erzeugung gemeinsam nutzbarer geheimer Schlüssel
  - Schlüsseltransportprotokolle: Sichere Verteilung eines Schlüssels
  - Schlüsselvereinbarungsprotokolle: Gemeinsame Erzeugung
  - Protokolle müssen Authentifizierung mit einbauen
- **Anonymität von Systemteilnehmern**
  - Senderanonymität: Protokolle für anonyme Hinweise
  - Empfängeranonymität: Protokolle für anonyme Anfragen (Chiffre)
  - Kommunikationsanonymität: Kommunikation nach außen unsichtbar

Zukunftsträchtige Themen

- **Hyperelliptische Kurven**
  - Verallgemeinerung elliptischer Kurven
  - Höhere Sicherheit durch größere Auswahl von Parametern
- **Quantenkryptographie**
  - Berechnungen auf der Basis von Quantenzuständen und -verschränkung
  - Mehr Effizienz, da Qubits beliebig viele Zustände annehmen können

    Anwendung: Effiziente Faktorisierung (Algorithmus von Shor)
  - Mehr Sicherheit, da Beobachtung von Daten nie unbemerkt bleibt

    z.B. Quantum-Key Distributuion, Quantum Direct Communication
- **Viele Anwendungen**
  - Internetsicherheit als wichtigstes Problem

    Maximale Freiheit bei größtmöglicher Sicherheit

    Leichte Anwendung bei maximalem Schutz naiver Benutzer