Vorlesung Sicherheit

Dennis Hofheinz

ITI, KIT

23.06.2014

Überblick

Überblick

Erinnerung Zero-Knowledge

- **Motivation:** PK-Identifikation, bei der
  1. V nichts über sk lernt
  2. P beweist, dass er sk kennt
- Zero-Knowledge-Eigenschaft (formalisiert 1): $\langle \mathrm{P}(sk), \mathcal{A}(pk) \rangle$-Transkript kann von Simulator $\mathcal{S}(pk)$ simuliert werden
- Signatur-/PKE-basierte PK-ID-Protokolle *nicht* ZK

Überblick

Nicht-Beispiel für ein ZK-Protokoll

- **Naheliegende Frage:** ist unser Protokoll

  1 $\mathrm{P} \xleftarrow{R} \mathrm{V}$

  2 $\mathrm{P} \xrightarrow{\sigma := \mathrm{Sig}(sk_A, R)} \mathrm{V}$

  Zero-Knowledge?
- **Antwort:** nein
  - Grund: schon für „ehrlichen“ Angreifer $\mathcal{A} = \mathrm{V}$ enthält Transkript $\langle \mathrm{P}(sk), \mathcal{A}(1^k, pk) \rangle$ eine gültige Signatur
  - Passender Simulator $\mathcal{S}$ müsste eine Signatur fälschen
  - Widerspruch zur EUF-CMA-Sicherheit des Signaturverfahrens
  - Wäre tatsächlich schon Widerspruch zur Sicherheit des PK-ID-Protokolls

Hilfsbaustein: Commitments

- Commitment-Verfahren besteht aus PPT-Algorithmus Com:
  - **Syntax:** $\mathrm{Com}(M; R)$ (mit Eingabe $M \in \{0,1\}^*$ und explizitem Zufall R für Ausführung, Sicherheitsparameter implizit)
  - **Ausgabe:** $com = \mathrm{Com}(M; R)$ ist Commitment auf M
- Intuition: *com* legt auf M fest
  - *com* kann nur für ein einziges M als $com = \mathrm{Com}(M; R)$ aufgedeckt werden
  - **Aber:** *com* soll M (noch) nicht verraten
  - Später kann *com* aufgedeckt werden, indem (M, R) veröffentlicht wird
- Beispiel: $\mathrm{Com}(M; R) = \mathrm{H}(M, R)$ für Hashfunktion H

Hilfsbaustein: Commitments

- Eigenschaften von Com formal(er):
  - **Hiding:** Für beliebige $M, M' \in \{0,1\}^*$ sind die Verteilungen

    $$\mathrm{Com}(M; R) \qquad \text{und} \qquad \mathrm{Com}(M'; R)$$

    ununterscheidbar (wobei R unabhängig zufällig)
  - **Binding:** Für jeden PPT-Angreifer $\mathcal{A}$ ist

    $$\Pr\left[\mathrm{Com}(M; R) = \mathrm{Com}(M', R') \wedge M \neq M'\right]$$

    vernachlässigbar in k, wobei die Wahrscheinlichkeit über $(M, R, M', R') \leftarrow \mathcal{A}(1^k)$ gemeint ist
- Beispiel ($\mathrm{Com}(M; R) = \mathrm{H}(M, R)$):
  - Binding folgt aus H-Kollisionsresistenz, Hiding nicht ganz klar

Beispiel für ein ZK-Protokoll

- **Erinnerung:** Graph-Dreifärbbarkeit
  - Geg.: Graph $G = (V, E)$ mit Knotenmenge $V = \{1, \ldots, n\}$ und Kantenmenge $E \subseteq V^2$
  - Eine *Dreifärbung* von G ist eine Abbildung $\phi : V \to \{1, 2, 3\}$ mit $(i, j) \in E \Rightarrow \phi(i) \neq \phi(j)$
  - G heißt *dreifärbbar*, wenn eine Dreifärbung von G existiert
  - Entscheidungsproblem (geg. G dreifärbbar?) NP-vollständig
- **Ziel:** Ein PK-Identifikationsprotokoll, dessen Sicherheit auf der Schwierigkeit, Dreifärbungen zu berechnen, beruht

Beispiel für ein ZK-Protokoll

- Gen wählt Graph G zusammen mit Dreifärbung ϕ und setzt

$$pk := G \qquad \text{und} \qquad sk := (G, \phi)$$

- Protokollablauf zwischen P und V:
  1. P wählt Bijektion $\pi : \{1,2,3\} \to \{1,2,3\}$ der Farben
  2. P committed sich mit $com_i = \mathrm{Com}(\pi(\phi(i)); R_i)$ auf dreigefärbten Graphen mit „vertauschten“ Farben
  3. P sendet alle Commitments $com_1, \ldots, com_n$ an V
  4. V wählt zufällige Kante $(i,j) \in E$ und sendet (i,j) an P
  5. P öffnet com_i, com_j (sendet $(\pi(\phi(i)), R_i), (\pi(\phi(j)), R_j)$ an V)
  6. V akzeptiert gdw. Openings gültig und $\pi(\phi(i)) \neq \pi(\phi(j))$

Beispiel für ein ZK-Protokoll

- **Korrektheit:** klar
- **Sicherheit (im PK-ID-Sinne):** so noch nicht
  - 1. Problem: P könnte „Glück haben“, dass Fast-Dreifärbung nicht auffliegt
  - Lösung: Protokoll k Mal durchführen
  - 2. Problem: Dreifärbungs-Suchproblem (Graph $\rightarrow$ Dreifärbung) je nach Verteilung nicht hard-on-average
  - Möglich: hard-on-average-Probleme auf Dreifärbung reduzieren
- **Zero-Knowledge:** ja! (Simulator $\mathcal{S}$ folgt)

Überblick

Beispiel für ein ZK-Protokoll

- **Benötigt:** Simulator $\mathcal{S}$, der $\langle \mathrm{P}(sk), \mathcal{A}(pk) \rangle$-Transkript liefert
  1. $\mathcal{S}$ spielt Protokoll mit $\mathcal{A}$, dabei übernimmt $\mathcal{S}$ Rolle von P
  2. $\mathcal{S}$ sendet $com_i = \mathrm{Com}(c_i, R_i)$ für zufällige $c_i \in \{1, 2, 3\}$ an $\mathcal{A}$
  3. Wenn $\mathcal{A}$ Kante $(i, j) \in E$ wählt, *hofft* $\mathcal{S}$ auf $c_i \neq c_j$
     - Wenn $c_i \neq c_j$: Protokoll fährt fort wie mit echtem P
     - Wenn $c_i = c_j$: $\mathcal{S}$ spult $\mathcal{A}$ zurück und spielt Spiel von neuem
  4. $\mathcal{S}$ gibt schließlich Transkript $\langle \mathcal{S}(pk), \mathcal{A}(pk) \rangle$ aus
- Einziger Unterschied zu echtem Transkript: $\mathcal{A}$ sieht in echtem Transkript „echte“ com_i, mit $\mathcal{S}$ aber „Zufalls-com_i“
  - Kein Problem: Wenn $\mathcal{A}$ Unterschied merkt, bricht er schon Hiding-Eigenschaft des Commitments
  - Formaler: Konstruiere Angreifer auf Hiding-Eigenschaft aus Unterscheider zwischen $\langle \mathrm{P}(sk), \mathcal{A}(pk) \rangle$ und $\langle \mathcal{S}(pk), \mathcal{A}(pk) \rangle$

Beispiel für ein ZK-Protokoll

- Einige Details zu Simulator $\mathcal{S}$:
  - $\mathcal{S}$ funktioniert auch, wenn mehrere Instanzen des Protokolls hintereinander ausgeführt werden (in dem Fall Rewinding von $\mathcal{A}$ zum Anfang der aktuellen Instanz)
  - Laufzeit von $\mathcal{S}$ nur *im Erwartungswert* polynomiell (polynomielle Laufzeit kann mit kleinem Fehler erkauft werden)
- Einige Details zu Protokoll:
  - Interpretation als interaktives Beweissystem möglich (interaktiver Beweis für Sprache aller dreifärbbaren Graphen)
  - V lernt dabei nichts über Dreifärbung, sondern nur, *dass* G dreifärbbar ist
  - Erlaubt, *beliebige* NP-Aussagen in Zero-Knowledge zu zeigen
- Mehr in weiterführenden Vorlesungen (Komplexitätstheorie)

Überblick

Proof-of-Knowledge-Eigenschaft

- **Erste Anforderung:** V lernt sk_A nicht ✓
- **Zweite Anforderung:** P kennt sk_A
- **Frage:** Wie könnte zweite Anforderung formalisiert werden?

Proof-of-Knowledge-Eigenschaft

Definition (Proof of Knowledge, informell)

Ein PK-Identifikationsprotokoll $(\mathrm{Gen}, \mathrm{P}, \mathrm{V})$ ist ein *Proof of Knowledge (POK)*, falls ein PPT-Algorithmus $\mathcal{E}$ (der „Extraktor“) existiert, so dass $\mathcal{E}^{\mathrm{P}'}(pk)$ bei Zugriff auf einen beliebigen erfolgreichen Prover P' schon einen geheimen Schlüssel *sk* zu *pk* extrahiert.

- Fehlende Details (in rot) können geeignet ausgefüllt werden
- **Intuition:** Wer als Prover überzeugt, kennt schon (ein) *sk*
- **Wichtig:** widerspricht nicht Zero-Knowledge! ($\mathcal{E}$ kann mehr als nur mit P' kommunizieren, z.B. auch P' rewinden)

Proof-of-Knowledge-Eigenschaft

- **Beispiel:** Das Graph-Dreifärbbarkeitsprotokoll
- **Ziel:** Einen Extraktor $\mathcal{E}$ definieren
- **Erinnerung:** Protokollablauf zwischen P und V
  1. P bietet V dreigefärbten Graph mit vertauschten Farben an
  2. V wählt eine Kante $(i, j) \in E$ aus
  3. P verrät V, welche Farben Knoten i und j haben
- **Idee:** (für P' als P und $\mathcal{E}$ in der Rolle von V)
  1. P' bietet $\mathcal{E}$ dreigefärbten Graph mit vertauschten Farben an
  2. $\mathcal{E}$ wählt eine Kante $(i, j) \in E$ aus
  3. P' verrät $\mathcal{E}$, welche Farben Knoten i und j haben
  4. $\mathcal{E}$ setzt P' auf Stand von 2 zurück und wählt *andere* Kante...
  5. ... bis $\mathcal{E}$ alle Knotenfarben kennt

Proof-of-Knowledge-Eigenschaft

- Einige Fragen und Antworten zu Extraktor
  - **Frage:** Was, wenn P' nur manche Kanten korrekt aufdeckt?
  - **Antwort:** Deshalb viele Wiederholungen von Protokoll
    - P' bei hinreichend vielen (etwa $|E| \cdot k$) Wiederholungen erfolgreich $\Rightarrow$ es gibt einen Durchlauf, bei dem P' alle Kanten aufdecken können muss
    - Wir untersuchen alle Durchläufe, bis wir diesen „guten“ Durchlauf finden und extrahieren dann alle Knotenfarben
  - **Frage:** Was ist mit den Knoten, die an keiner Kante hängen?
  - **Antwort:** Deren Farbe kann beliebig gesetzt werden

Überblick

Zusammenhang mit Identifikationssicherheit

- **Frage:** Ist jedes Identifikationsprotokoll, das ZK- und POK-Eigenschaft hat, als Identifikationsprotokoll sicher?
- **Antwort:** Nicht notwendig, z.B. könnten Dreifärbungen einfach zu finden sein (je nach Verteilung von G)
- **Aber:** *Wenn* $pk \rightarrow sk$-Problem hard-on-average ist, und Protokoll ZK- und POK-Eigenschaften hat, *dann* gilt Sicherheit im Identifikationssinne
- Kein Beweis, aber Intuition hierfür (Reduktion):
  - Wegen ZK-Eigenschaft lernt Angreifer $\mathcal{A}$ nichts in Phase 1
  - Aus erfolgreichem Angreifer $\mathcal{A}$ kann sk extrahiert werden
  - Aus erfolgreichem $\mathcal{A}$ lässt sich so ein erfolgreicher $pk \rightarrow sk$-Löser konstruieren

Überblick

Weitere Anwendungen von Zero-Knowledge

- **Beobachtung:** Graph-Dreifärbbarkeitsprotokoll lässt sich als (interaktives) Beweissystem für NP interpretieren
  - P beweist V, *dass* G dreifärbbar (ohne Zeugen zu verraten)
  - Sprache aller dreifärbbaren Graphen NP-vollständig
  - $\Rightarrow$ *beliebige* NP-Aussagen in ZK beweisbar
- **Beispiel:** P kann V folgende Aussage (zu gegebenen Public Keys und gegebenem Chiffrat C) beweisen:

  > *Dieses Chiffrat enthält eine gültige Signatur zur Nachricht „*P *ist* X *Jahre alt“ für* $X \geq 18$.

  ohne dabei X preiszugeben
- **Beispielanwendung:** Credential Systems (elektronische ID)

Überblick

Zusammenfassung Zero-Knowledge

- Nützlich zu Identifikationszwecken, generell extrem mächtig
  - Zero-Knowledge: V lernt *nichts*
  - Proof of Knowledge: P muss *sk* kennen
- Beispiel: Graph-Dreifärbbarkeitsprotokoll
- Allgemeiner: NP-Aussagen zeigen, ohne Zeugen herzugeben
- Anwendung als Baustein in größeren Protokollen:
  - Trusted Platform Module (TPM)
  - Anonymous Credential Systems
  - Elektronische Wahlverfahren
  - Mehrparteienberechnungen

Aktuelle Forschung Zero-Knowledge

- Effiziente Zero-Knowledge-Protokolle
  - Gut verstanden: über zyklischen Gruppen
  - Weniger gut verstanden: über Gittern
- Nicht-interaktive Zero-Knowledge-Protokolle
  - Neue Anwendungen (Verschlüsselungssysteme)
  - Mit Pairings effizient möglich
  - Ohne Pairings (Gitter, RSA-Szenario) nicht gut verstanden
- Praktikable Anonymous Credential Systems
- Elektronische Wahlverfahren
  - Wählern Korrektheit der Auszählung beweisen
  - Weitergabe/Wiederverwendung von Stimmen verhindern