

Vorlesung Sicherheit

Dennis Hofheinz

ITI, KIT

26.06.2017

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

1 Identifikationsprotokolle

- Erinnerung
 - Sicherheitsmodell
 - Ein sicheres Protokoll
 - Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

- Ziel: asymmetrische Authentifikation (von Parteien)

$$P_{sk_A} \longleftrightarrow V$$

- pk_A öffentlich
- Nicht-interaktive Protokolle unsicher

1 Identifikationsprotokolle

- Erinnerung
- **Sicherheitsmodell**
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

Formalisierung PK-Identifikationsprotokoll

- (Public-Key-)Identifikationsprotokoll: (Gen, P, V)
- PPT-Algorithmus $\text{Gen}(1^k)$ gibt Schlüsselpaar (pk, sk) aus
- Zwei PPT-Algorithmen P, V mit *Zustand* interagieren:
 - 1 V wird mit Eingabe pk gestartet, Ausgabe sei out_V
 - 2 P wird mit Eingaben sk und out_V gestartet, Ausgabe out_P
 - 3 V wird mit Eingabe out_P gestartet, Ausgabe out_V
 - Ist $\text{out}_V \in \{0, 1\}$, dann beende die Interaktion
 - Andernfalls zurück zu Schritt 2 (sk -Eingabe nicht mehr nötig)
- **Notation:** $\langle P(sk), V(pk) \rangle$ ist Transkript der Interaktion
- **Korrektheit:** V gibt schließlich 1 aus für $(pk, sk) \leftarrow \text{Gen}(1^k)$

Sicherheit eines PK-Identifikationsprotokolls

- PK-ID-Protokoll (Gen, P, V) sicher $:\Leftrightarrow$ kein PPT-Angreifer \mathcal{A} gewinnt folgendes Spiel mehr als vernachlässigbar oft:
 - **Phase 1:** \mathcal{A} darf mit beliebig vielen P -Instanzen (mit sk_i) in der Rolle des Verifiers V (mit Eingabe pk_i) interagieren. Die verwendeten $(pk_i, sk_i) \leftarrow \text{Gen}(1^k)$ sind vom Spiel gewählt.
 - **Phase 2:** \mathcal{A} sucht sich ein schon vom Spiel gewähltes pk_{i^*} aus und interagiert mit einer V -Instanz (mit Eingabe pk_{i^*})
 - **Entscheidung:** \mathcal{A} gewinnt, wenn V schließlich 1 ausgibt
- **Intuition:** Kein \mathcal{A} schafft es, andere zu impersonieren
- **Allerdings:** Verhindert keinen Man-in-the-Middle-Angriff

1 Identifikationsprotokolle

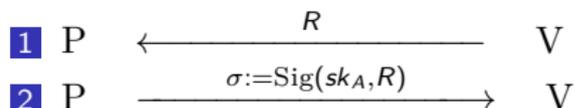
- Erinnerung
- Sicherheitsmodell
- **Ein sicheres Protokoll**
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

Ein sicheres PK-Identifikationsprotokoll

■ Erinnerung Kandidat (Gen, P, V)



Theorem (Sicherheit von (Gen, P, V))

Ist das verwendete Signaturverfahren EUF-CMA-sicher, so ist das obige PK-Identifikationsprotokoll (Gen, P, V) sicher.

Beweisidee.

Konstruiere EUF-CMA-Angreifer \mathcal{B} aus PK-ID-Angreifer \mathcal{A} . □

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- **Noch ein sicheres Protokoll**

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

Noch ein sicheres PK-Identifikationsprotokoll

- Ähnlicher Kandidat (Gen, P, V) mit Verschlüsselung:

$$\begin{array}{l} \text{1 } P \xleftarrow{C \leftarrow \text{Enc}(pk_A, R)} V \\ \text{2 } P \xrightarrow{R = \text{Dec}(sk_A, C)} V \end{array}$$

Theorem (Sicherheit von (Gen, P, V))

Ist das verwendete Verschlüsselungsverfahren IND-CCA-sicher¹, so ist das obige PK-Identifikationsprotokoll (Gen, P, V) sicher.

- Beweisidee wie im signaturbasierten Protokoll
- **Achtung:** (pk_A, sk_A) nicht auch zur Verschlüsselung benutzen

¹ „[Ciphertext] Indistinguishability under Chosen-Ciphertext Attacks“, wie IND-CPA, modelliert aber aktive Angriffe: \mathcal{A} erhält $\text{Dec}(sk, \cdot)$ -Orakel

- Identifikation notwendig interaktiv
- Signatursysteme (oder aktiv sichere PKE-Verfahren) hinreichend für sichere Identifikation
 - Ähnliche einfache Identifikationsprotokolle implizit z.B. in TLS verwendet
- Nicht behandelt: stärkere Sicherheitsbegriffe (Man-in-the-Middle-Angriffe, Reset-Angriffe, ...)

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- **Motivation**
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

Motivation Zero-Knowledge

- Einige intuitive (naive?) Anforderungen an PK-ID-Protokoll noch nicht oder nur teilweise erfüllt:

1 V lernt sk_A nicht

2 V ist sicher, dass Gegenüber sk_A kennt

- Erinnerung Identifikation mit Signaturen:

1 $P \xleftarrow{R} V$

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

- Zwar lernt V nicht den kompletten sk_A . . .
... aber vielleicht Teilinformationen über sk_A
- Vielleicht kennt P nur „Ersatz“- sk_A
- **Frage:** Für Identifikation beides nicht schlimm. . .
... können wir trotzdem intuitive Anforderungen erfüllen?

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- **Zero-Knowledge-Eigenschaft**
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

Zero-Knowledge-Eigenschaft

- **Erste Anforderung:** V lernt sk_A nicht
- **Keine halben Sachen:** V lernt nichts über sk_A
- **Zurückrudern:** V lernt nichts über sk_A , was er nicht schon aus pk_A berechnen kann (Bsp.: $sk_A = x$, $pk_A = g^x$)
- **Anders gesagt:** Alles, was V über sk_A berechnen kann, kann er schon aus pk_A berechnen
- **Randbedingung:** Natürlich muss das auch für „bösen V “ (d.h. für Angreifer \mathcal{A} in der Rolle von V) gelten

- Hilfsformalismus: Ununterscheidbarkeit

Definition (Ununterscheidbarkeit)

Zwei (möglicherweise vom Sicherheitsparameter $k \in \mathbb{N}$ abhängige) Verteilungen X, Y sind *ununterscheidbar* (geschrieben $X \stackrel{c}{\approx} Y$), wenn für alle PPT-Algorithmen \mathcal{A} die Differenz

$$\Pr \left[\mathcal{A}(1^k, x) = 1 \mid x \leftarrow X \right] - \Pr \left[\mathcal{A}(1^k, y) = 1 \mid y \leftarrow Y \right]$$

vernachlässigbar in k ist.

- **Intuition:** X und Y nicht (effizient) unterscheidbar

Zero-Knowledge-Eigenschaft (formal)

Definition (Zero-Knowledge)

Ein PK-Identifikationsprotokoll (Gen, P, V) ist *Zero-Knowledge* (ZK), falls für jeden PPT-Algorithmus \mathcal{A} (den Angreifer) ein PPT-Algorithmus \mathcal{S} (der Simulator) existiert, so dass die folgenden Verteilungen ununterscheidbar sind (wobei $(pk, sk) \leftarrow \text{Gen}(1^k)$):

$$\left(pk, \langle P(sk), \mathcal{A}(1^k, pk) \rangle \right) \quad \text{und} \quad \left(pk, \mathcal{S}(1^k, pk) \right).$$

- **Intuition:** Interaktionstranskripte simulierbar
- **Bemerkung:** \mathcal{A} kann ganzes Wissen in Transkript packen
- Varianten möglich (z.B. Gleichheit statt Ununterscheidbarkeit)

1 Identifikationsprotokolle

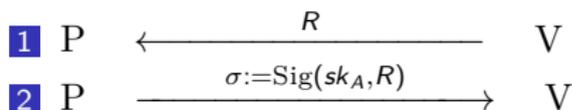
- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- **Beispiel für Zero-Knowledge-Protokoll**
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

Nicht-Beispiel für ein ZK-Protokoll

- **Naheliegende Frage:** ist unser Protokoll



Zero-Knowledge?

- **Antwort:** nein
 - Grund: schon für „ehrlichen“ Angreifer $\mathcal{A} = V$ enthält Transkript $\langle P(sk), \mathcal{A}(1^k, pk) \rangle$ 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:** $\text{Com}(M; R)$ (mit Eingabe $M \in \{0, 1\}^*$ und explizitem Zufall R für Ausführung, Sicherheitsparameter implizit)
 - **Ausgabe:** $\text{com} = \text{Com}(M; R)$ ist Commitment auf M
- Intuition: com legt auf M fest
 - com kann nur für ein einziges M als $\text{com} = \text{Com}(M; R)$ aufgedeckt werden
 - **Aber:** com soll M (noch) nicht verraten
 - Später kann com aufgedeckt werden, indem (M, R) veröffentlicht wird
- Beispiel: $\text{Com}(M; R) = H(M, R)$ für Hashfunktion H

Hilfsbaustein: Commitments

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

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

ununterscheidbar (wobei R unabhängig zufällig)

- **Binding:** Für jeden PPT-Angreifer \mathcal{A} ist

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

vernachlässigbar in k , wobei die Wahrscheinlichkeit über $(M, R, M', R') \leftarrow \mathcal{A}(1^k)$ gemeint ist

- Beispiel ($\text{Com}(M; R) = \text{H}(M, R)$):
 - Binding folgt aus H-Kollisionsresistenz, Hiding nicht ganz klar

- **Erinnerung:** Graph-Dreifärbbarkeit
 - Geg.: Graph $G = (V, E)$ mit Knotenmenge $V = \{1, \dots, n\}$ und Kantenmenge $E \subseteq V^2$
 - Eine *Dreifärbung* von G ist eine Abbildung $\phi : V \rightarrow \{1, 2, 3\}$ 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

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

- Protokollablauf zwischen P und V:

- 1 P wählt Bijektion $\pi : \{1, 2, 3\} \rightarrow \{1, 2, 3\}$ der Farben
- 2 P committed sich mit $com_i = \text{Com}(\pi(\phi(i)); R_i)$ auf dreigefärbten Graphen mit „vertauschten“ Farben
- 3 P sendet alle Commitments com_1, \dots, 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)

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- **Analyse des Beispiel-Zero-Knowledge-Protokolls**
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

Beispiel für ein ZK-Protokoll

- **Benötigt:** Simulator \mathcal{S} , der $\langle P(sk), \mathcal{A}(pk) \rangle$ -Transkript liefert
 - 1 \mathcal{S} spielt Protokoll mit \mathcal{A} , dabei übernimmt \mathcal{S} Rolle von P
 - 2 \mathcal{S} sendet $com_i = \text{Com}(c_i, R_i)$ 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 $\langle P(sk), \mathcal{A}(pk) \rangle$ und $\langle \mathcal{S}(pk), \mathcal{A}(pk) \rangle$

Beispiel für ein ZK-Protokoll

- Einige Details zu Simulator S :
 - 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 S nur *im Erwartungswert* polynomiell (polynomielle Laufzeit kann mit kleinem Fehler erkaufte 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)

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- **Proof-of-Knowledge-Eigenschaft**
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

Proof-of-Knowledge-Eigenschaft

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

Definition (Proof of Knowledge, informell)

Ein PK-Identifikationsprotokoll (Gen, P, V) ist ein *Proof of Knowledge (POK)*, falls ein PPT-Algorithmus \mathcal{E} (der „Extraktor“) existiert, so dass $\mathcal{E}^{P'}(pk)$ 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

- 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

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- **Beziehung zu Identifikationssicherheit**
- Weitere Anwendungen
- Zusammenfassung

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 (verstärkte) 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

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- **Weitere Anwendungen**
- Zusammenfassung

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 Chiffirat C) beweisen:

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

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

1 Identifikationsprotokolle

- Erinnerung
- Sicherheitsmodell
- Ein sicheres Protokoll
- Noch ein sicheres Protokoll

2 Zero-Knowledge-Protokolle

- Motivation
- Zero-Knowledge-Eigenschaft
- Beispiel für Zero-Knowledge-Protokoll
- Analyse des Beispiel-Zero-Knowledge-Protokolls
- Proof-of-Knowledge-Eigenschaft
- Beziehung zu Identifikationssicherheit
- Weitere Anwendungen
- Zusammenfassung

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

- 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