Algorithmische Kryptographie (WS2012/13)

Kapitel 6
Einfache Protokolle

Walter Unger

Lehrstuhl für Informatik 1

18:41 Uhr, den 12. Dezember 2012

Inhalt I

Einleitung

Definition

Ein Protokoll ist der Ablauf eines Informationsaustausches, um eine spezielle Information zu übertragen oder zu generieren. Dabei ist diese spezielle Information vorher bei keinem Partner vorhanden.

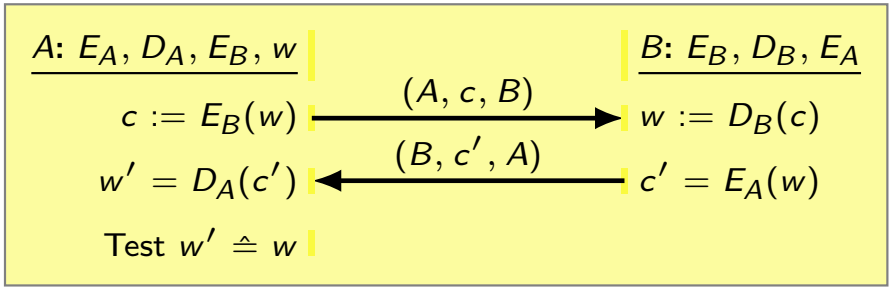
Folgende Fälle können auftreten:

- Es kann eine vertrauenswürdige Partei geben. Das ist der einfache, nicht realistische Fall.
- Partner können sich nicht vertrauen.
- Weitere Personen können lauschen.
- Weitere Personen können aktiv in das Protokoll eingreifen („aktives Lauschen").

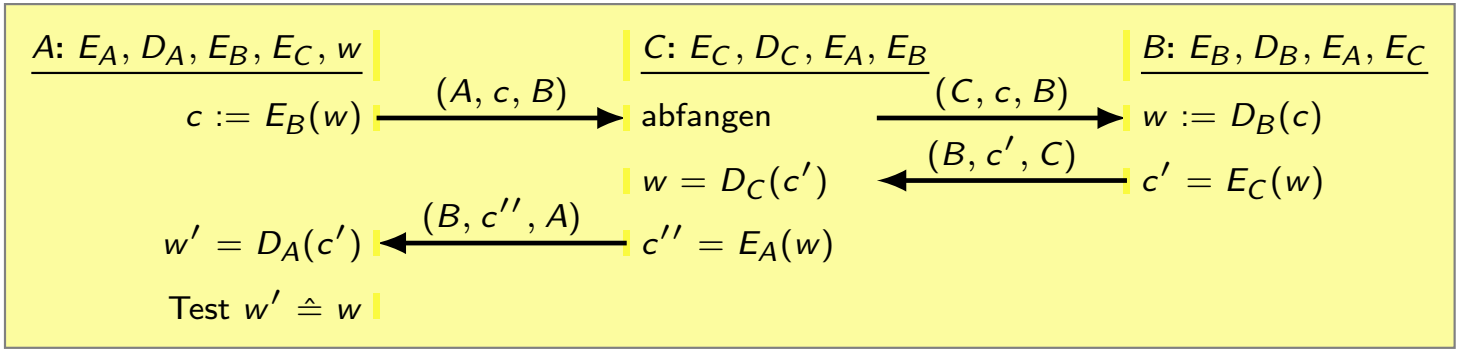
- Schlüsselaustausch
- Quittungen, Unterschriften, Identifikation, Gruppenunterschriften
- Gemeinsame Informationen bestimmen, würfeln einer Zahl
- Information kaufen, ohne dass der Verkäufer weiß, was gekauft wird
- Unsicher übertragen
- Wahlen, Elektronisches Geld

Sehr einfache Quittung

- Quittungen sind Empfangsbestätigungen des Protokollpartners.



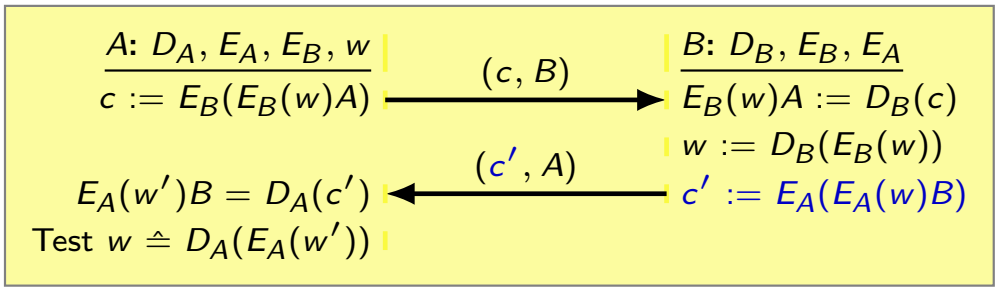
- Dieses Protokoll kann durch aktives Lauschen entschlüsselt werden.
- Aktiver Lauscher C, ein legaler Nutzer des Systems:



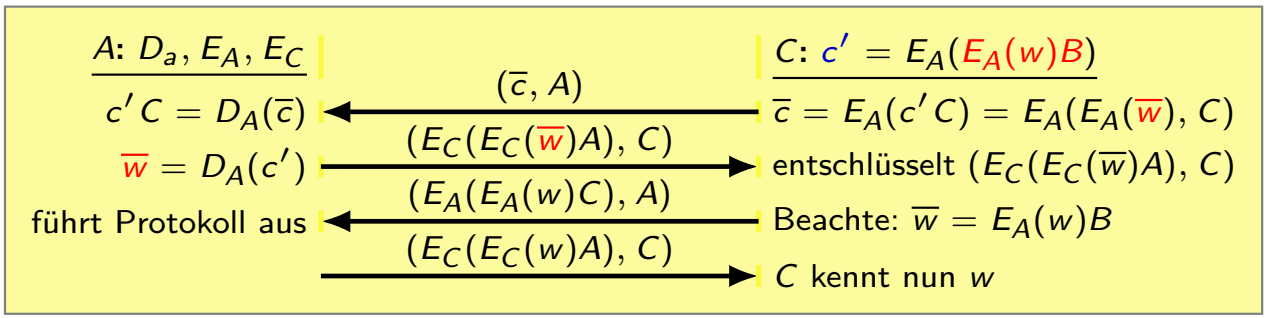
- Die Schwachstelle ist die unverschlüsselte Übermittlung des Absenders.

Einfache Quittungen

- Geben wir nun eine Verbesserung an (sende: $E_X(E_X(w)Y)$):



- Dieses Protokoll kann durch einen passiven Lauscher entschlüsselt werden.
- Dabei wird folgende Idee benutzt: Sende Nachricht $\overline{w} = E_A(w)B$ an A.
- Nach oben genannten Protokoll startet C wie folgt:

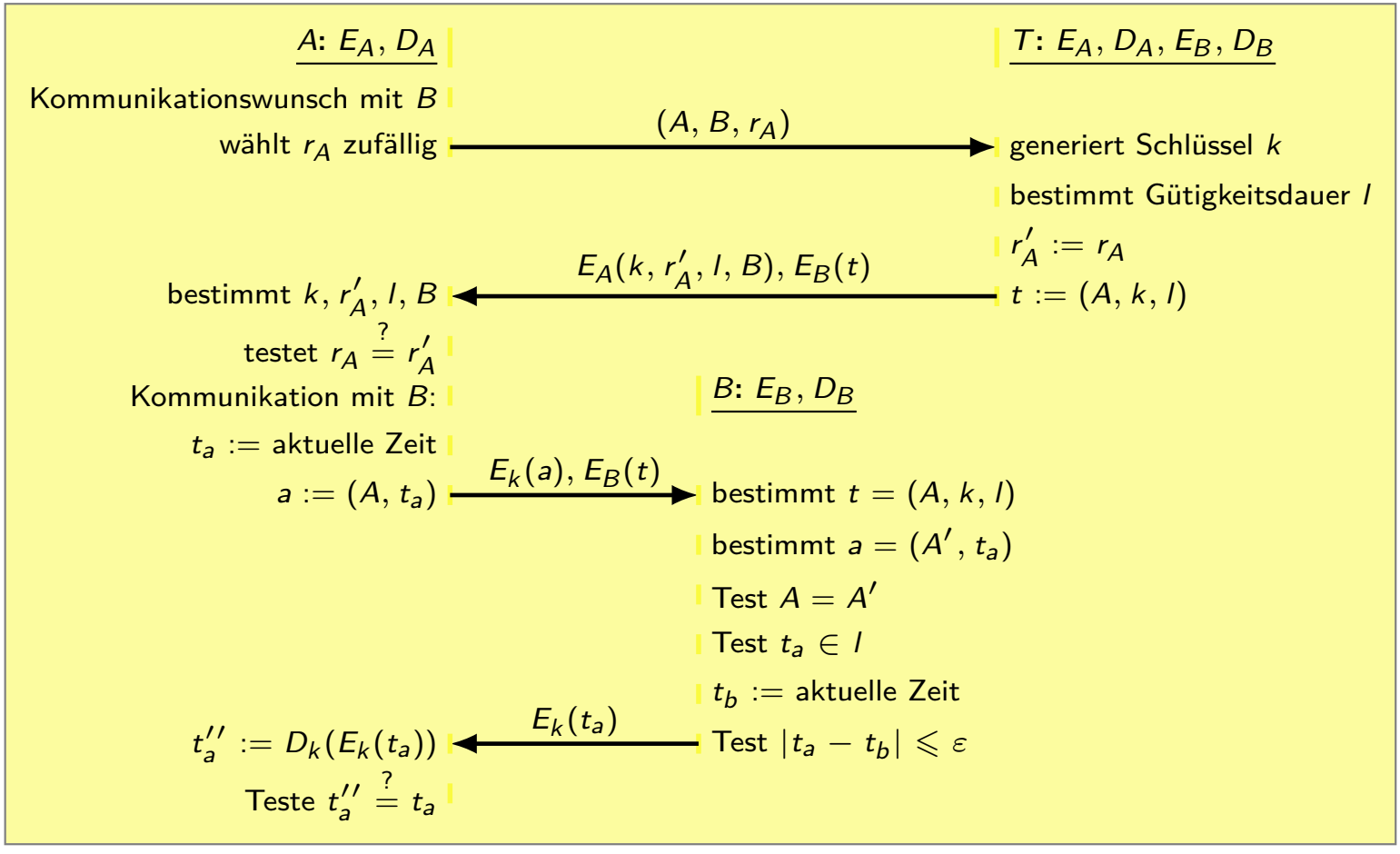


- Gegenmaßnahmen:
  - Speichere alle gesendeten Nachrichten oder nutze Hashwerte.
  - Verwende zwei Verschlüsselungen E_A und E'_A ($E_A(E'_A(w)B)$).

Kerberos-Verfahren

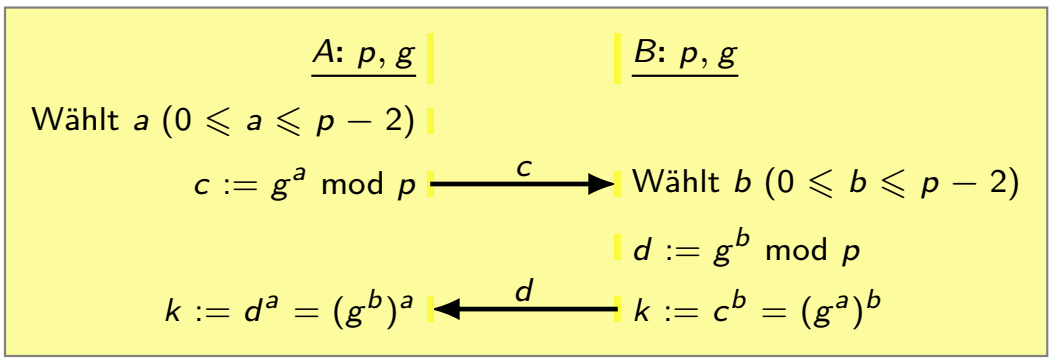
- Verfahren beruht auf einem symmetrischen Verschlüsselungsverfahren.
- Beteiligt sind A, B, T.
- T hat für jeden der Partner $A, B, \ldots$ einen Schlüssel für ein symmetrisches Verschlüsselungsverfahren.
- Verfahren im Überblick:
  - A will eine Kommunikation mit B beginnen und fragt bei einem vertrauenswürdigen Zentrum T nach einem Schlüssel (einem Session-Key) für die Kommunikation mit B.
  - A bekommt einen Schlüssel k und eine Information $E_B(t)$, mit der er gegenüber B belegen kann, dass er diese Information nur von T bekommen hat und damit ein legimitierter Teilnehmer ist.
  - Die Information $E_B(t)$ beinhaltet Zeitstempel.
  - Die Uhren müssen relativ genau gehen, und das von T erhaltene Ticket $E_B(t)$ kann innerhalb einer Zeitspanne von l mehrfach verwendet werden.

Kerberos-Verfahren



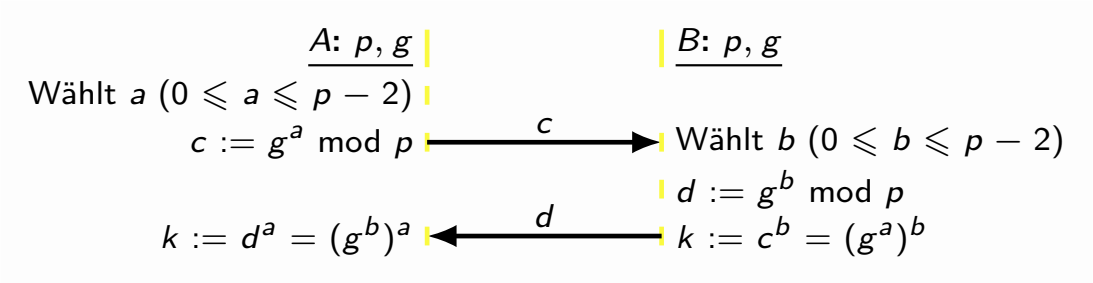
Diffie-Hellman Schlüsselaustausch

- Öffentlich bekannt sind
  - große Primzahl p und
  - Generator g in $\mathbb{Z}_p^*$.
- Damit können A und B einen gemeinsamen Schlüssel bestimmen.



Bemerkungen

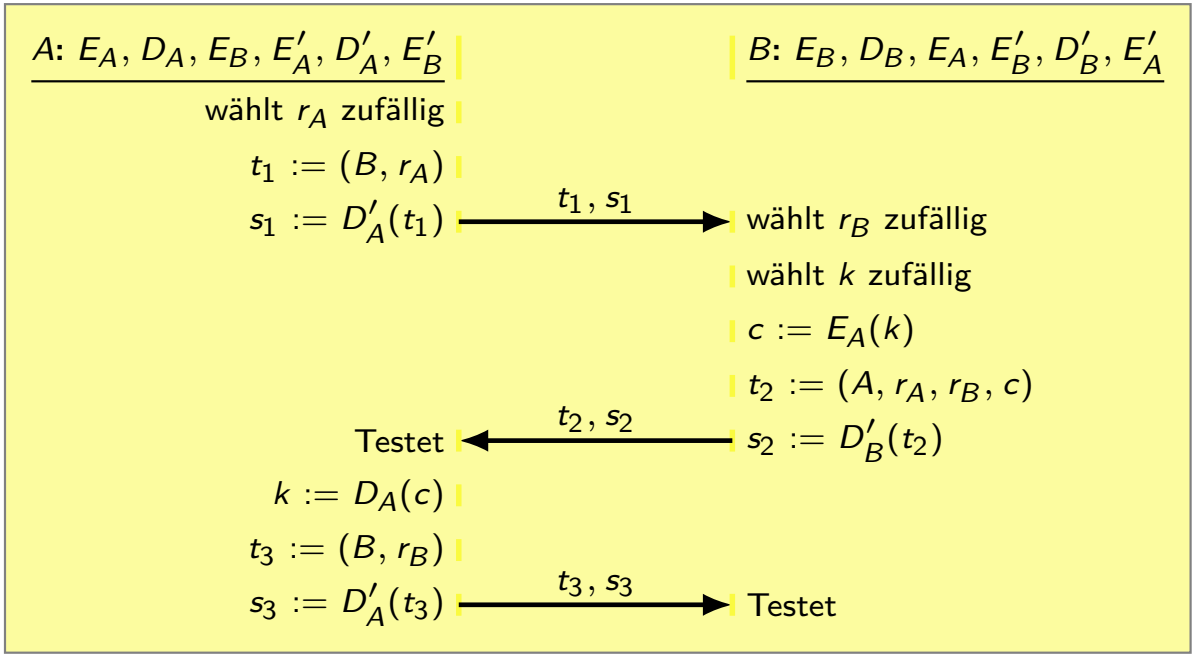
g^a, g^b, g^{ab}



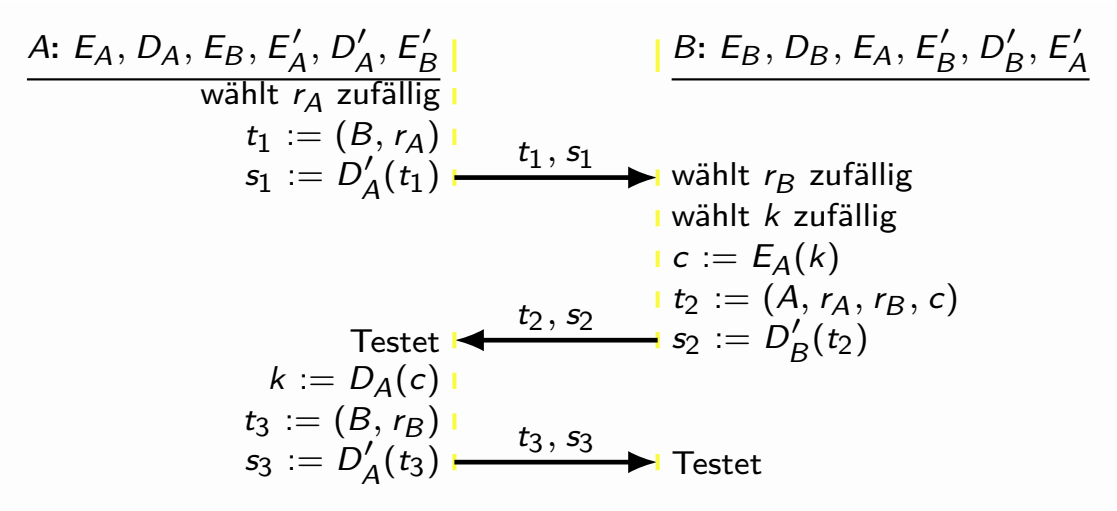
1. Falls jemand den diskreten Logarithmus berechnen kann, so kann er k aus d und c bestimmen. D.h. aus c bestimmt er a und setzt $k := d^a$.
2. Ansonsten ist g^{ab} aus g^a und g^b zu bestimmen. Dies ist das Diffie-Hellman Problem, welches auch als schwer betrachtet wird.
3. Nicht alle Bits von k sind beweisbar sicher. Für p werden 1024 Bits empfohlen. Da aber typischerweise dann k für ein symmetrisches Verfahren verwendet wird, hat k 128 Bits. Davon sind dann nur $\sqrt{p}$ d.h. 32 Bits sicher. Daher wird die Verwendung einer zusätzlichen Hashfunktion empfohlen.
4. Das Verfahren ist sicher bei einem passiven Lauscher, aber nicht bei einem aktiven Lauscher.

Schlüsselaustausch und Authentifizierung

- Hier ergibt sich ein 3-Runden Protokoll.
- Verwendet werden von jedem Partner ein Public-Key-Verfahren zur Verschlüsselung und eines zur Authentifizierung.



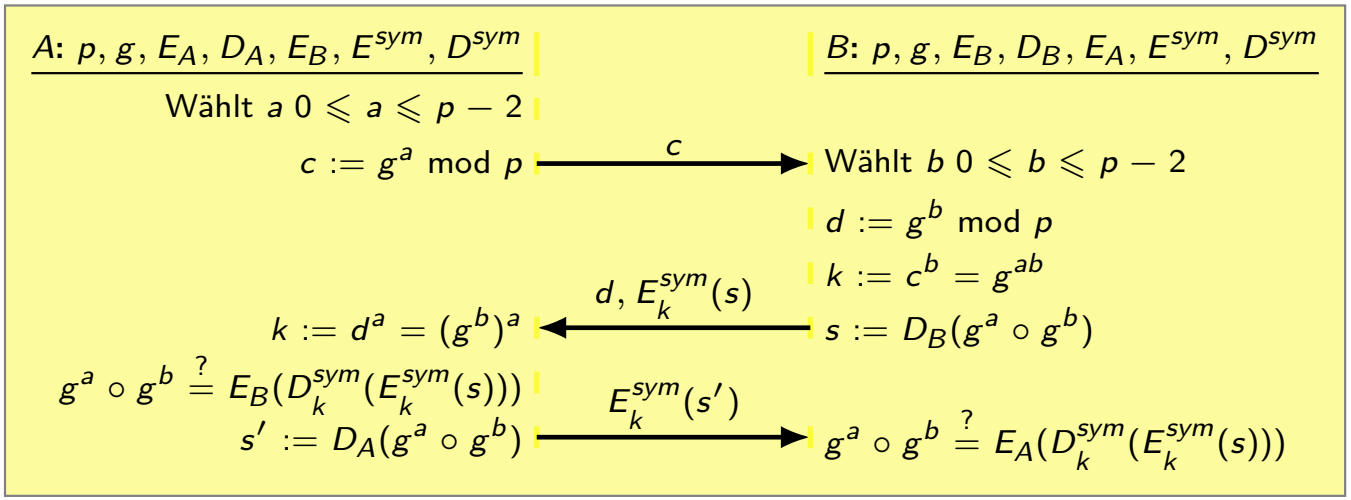
Bemerkungen:



1. Die Idee bei dem Protokoll ist ein Challenge–Response, d.h. es wird eine Anfrage gestellt, die nur der richtige Partner beantworten kann.
2. Das Protokoll kann auch in zwei Runden durchgeführt werden. Aber dann sind aus Sicherheitsgründen alle Nachrichten zu speichern.
3. Falls jemand alle Daten speichert und später einen Schlüssel von X bestimmen kann, so kann er auch jede Kommunikation von X entschlüsseln.
4. Dieses kann man verhindern, indem man dieses Verfahren mit Diffie-Hellman Schlüsselaustausch kombiniert.

Station to Station Schlüsselaustausch

- Hier wird Diffie-Hellman Schlüsselaustausch mit Authentifizierung verbunden.
- Öffentlich bekannt sind große Primzahl p und Generator g in $\mathbb{Z}_p^*$.
- Damit können A und B einen gemeinsamen Schlüssel bestimmen und sich gegenseitig authentifizieren.
- Weiterhin sei E^{sym}, D^{sym} ein symmetrisches Verschlüsselungsverfahren.

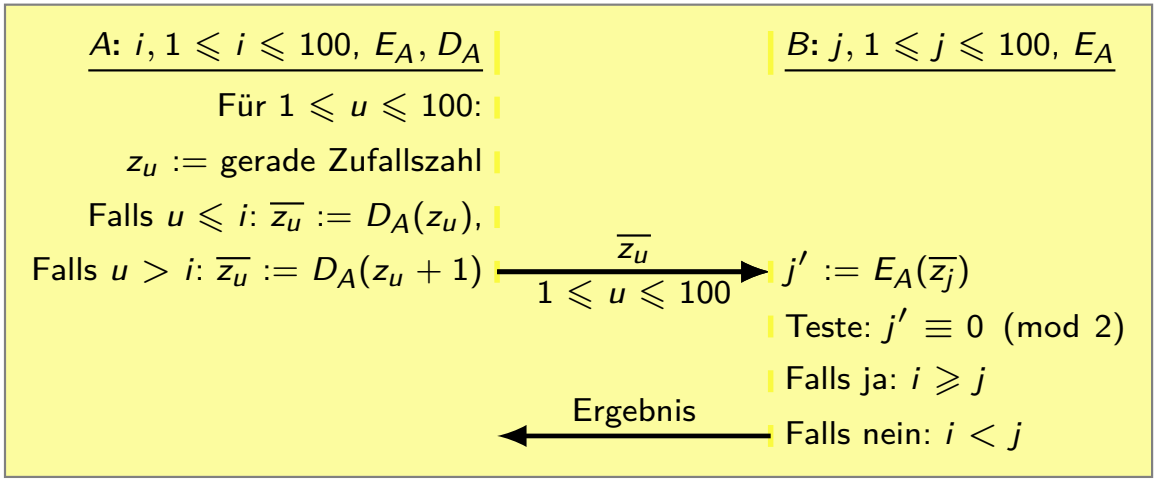


Gemeinsame Information

- Die Bestimmung gemeinsamer Information, die im folgenden protokollarisch durchgeführt wird, hat immer zum Ziel, die Grundlage der Information zu schützen, d.h. nicht zu verraten.
- Beispielsweise will man bei der Bestimmung, welche von zwei Personen älter ist, nicht das genaue Alter der Personen preisgeben.
- Ziel des folgenden Protokolls ist herauszufinden, ob $i < j$ gilt. i und j sollen dem jeweils anderen Protokollpartner aber nicht verraten werden.
- Anschauliches Beispiel:
  - A kennt i und B kennt j. Ziel: bestimme $i < j$.
  - A bereit 100 Umschläge U_x vor, in U_x mit $x < i$ kommt Zettel.
  - B öffnet U_j. Falls U_j keinen Zettel enthällt, gilt: $i < j$.

Gemeinsame Information (Idee)

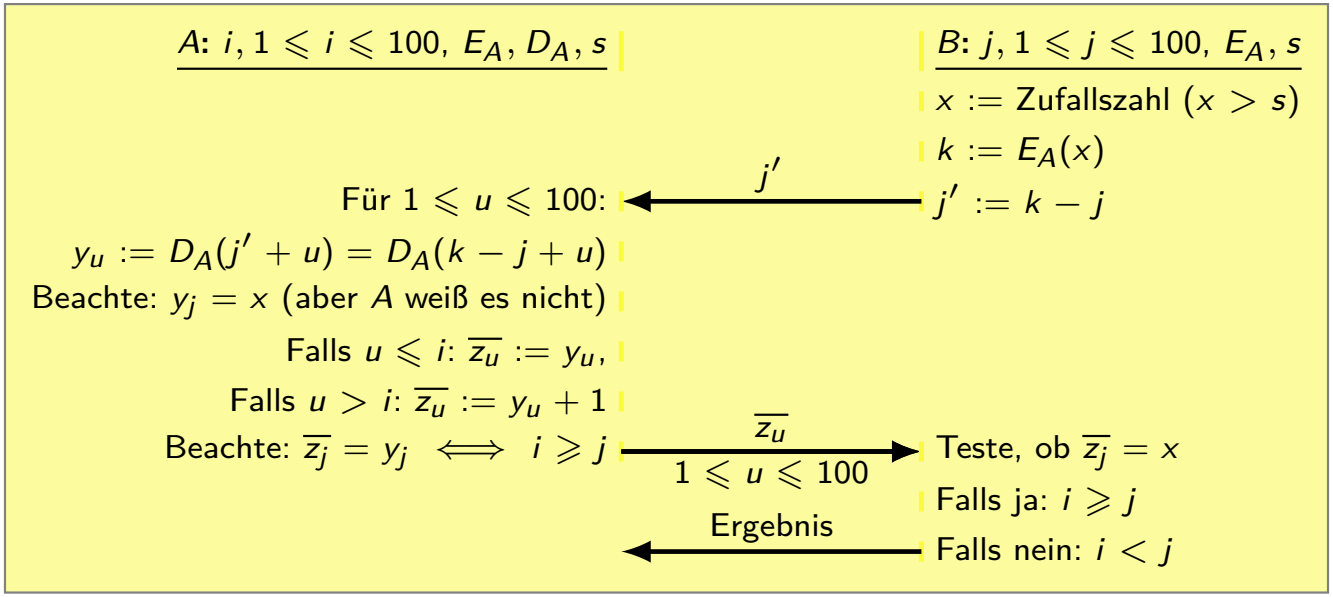
- Ein erster Versuch:



- Problem: B kann alle Zahlen $\overline{z_j}$ untersuchen.
- Problem: B kann vor dem letzten Schritt das Protokoll abbrechen.
- Das letzte Problem kann man nicht verhindern.

Gemeinsame Information

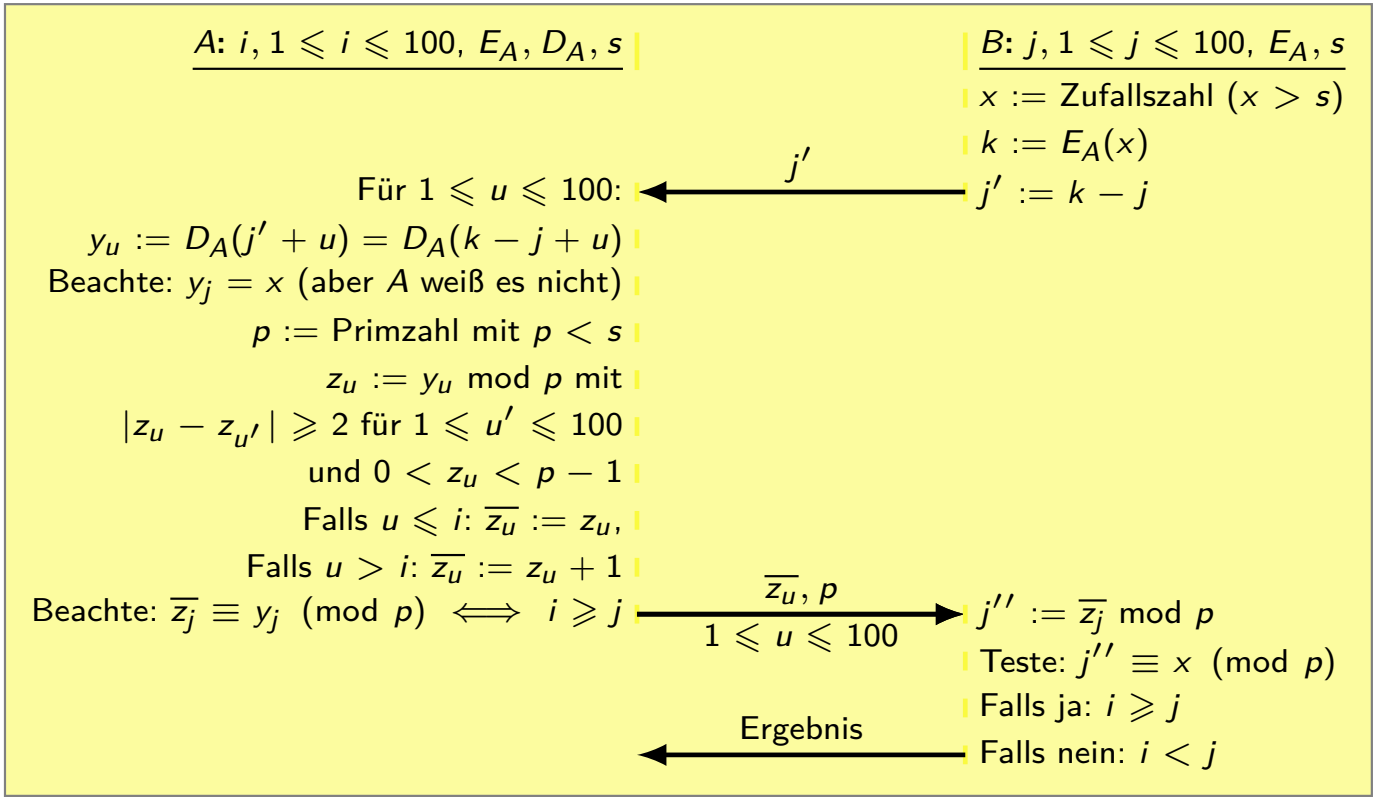
- B kodiert seinen Wert j für A.
- Ein zweiter Versuch:



- Es werden die Werte y_u und $y_u + 1$ geschickt.
- Damit kann B mit Hilfe von E_A die ursprünglichen Werte bestimmen und damit das Alter von A.

Gemeinsame Information

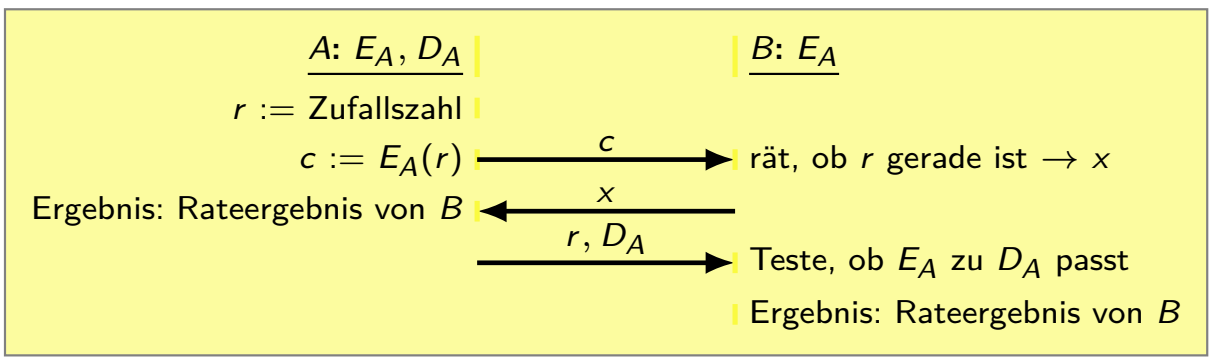
- Das finale Protokoll:



- Es wird keine weitere Information zwischen den Parteien ausgetauscht.
- Beim Alter kann gelogen werden.
- E_A, D_A dürfen nur einmal verwendet werden.

Einleitung

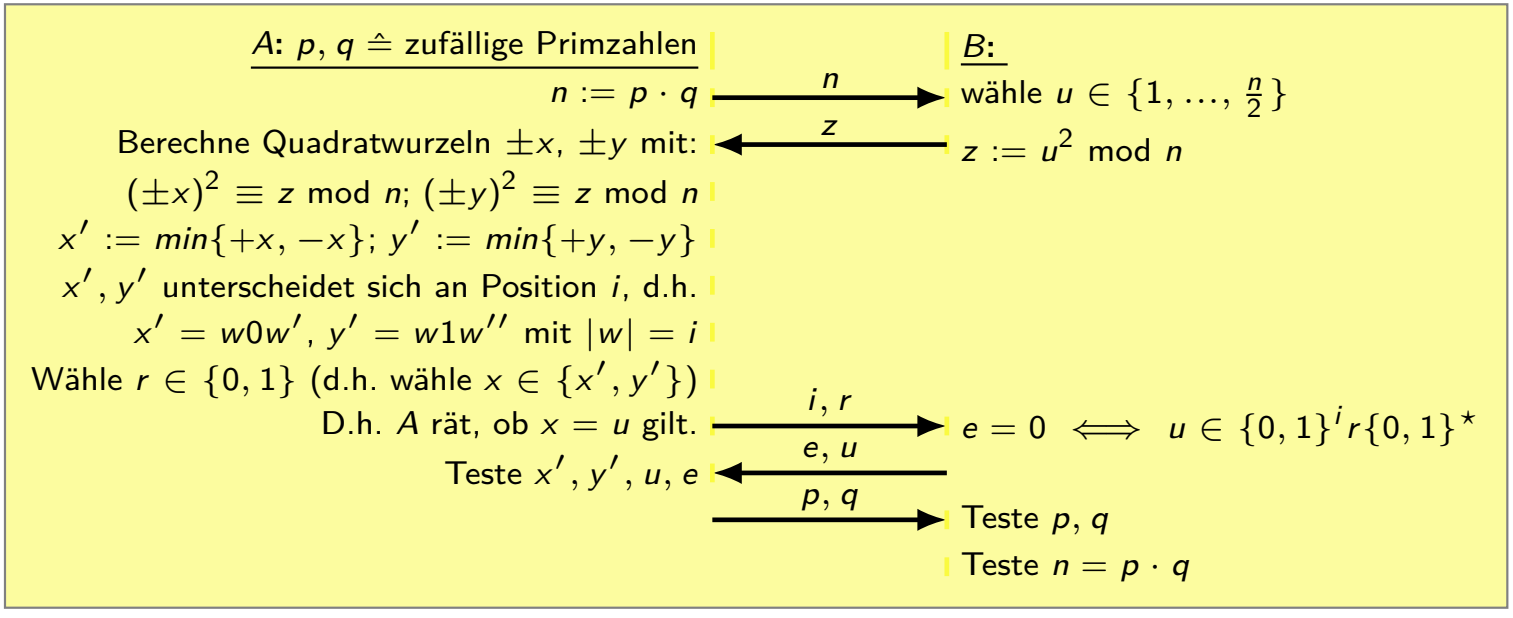
- Erzeugen einer Zufallszahl ohne vertrauenswürdigen Partner.
- Eine andere Formulierung dieses Problems wäre das Spielen von Skat über das Internet. Idee:
  1. A wirft eine Münze.
  2. B rät daraufhin das Ergebnis, A zeigt die Münze, ohne zu manipulieren (A kommt selbst an die Münze nicht mehr heran).
- Einfaches Protokoll:



- A sollte zufällig raten. Tut er dies nicht, so setzt er sich der Gefahr aus, dass B seine Strategie erkennt und dementsprechend trickreich das Ergebnis des Münzwurfs rät.

Verfahren mittels Quadratwurzeln

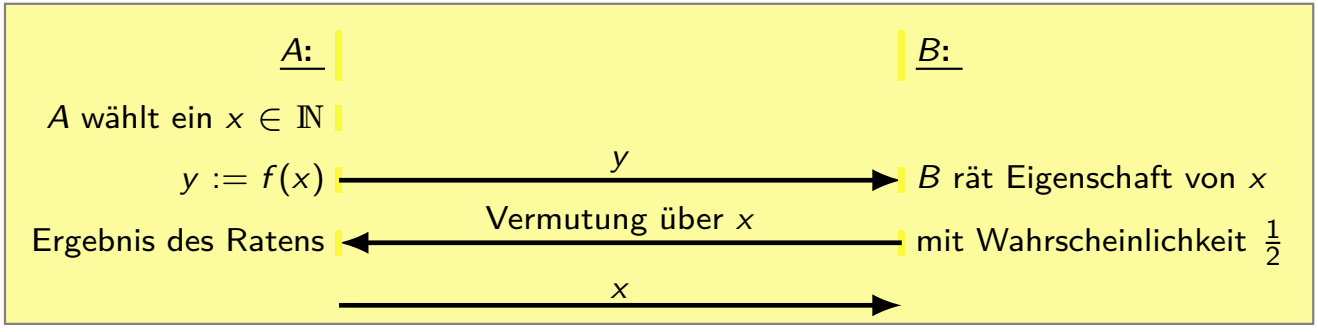
- Idee: Rate Eigenschaft einer Quadratwurzel:



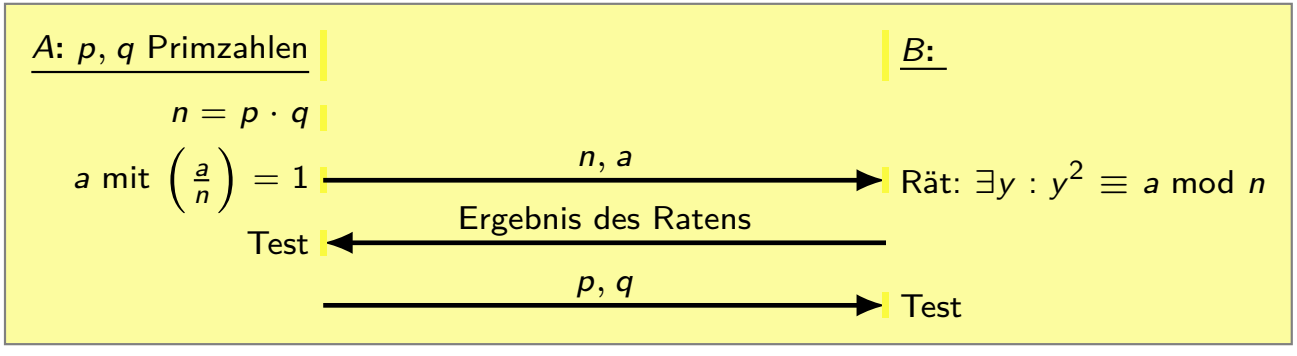
- A kann u nicht kennen. Also ist das Raten echt.
- B kann x', y' nicht kennen, denn sonst kann B faktorisieren und
- $\text{ggT}(x' - y', n) \in \{p, q\} \Rightarrow B$ kann u nicht ändern
- Beachte: Falls A anstelle eines Bits r den Wert x' oder y' komplett sendet, so könnte B ggf. das Ergebnis verändern!

Allgemeines Vorgehen

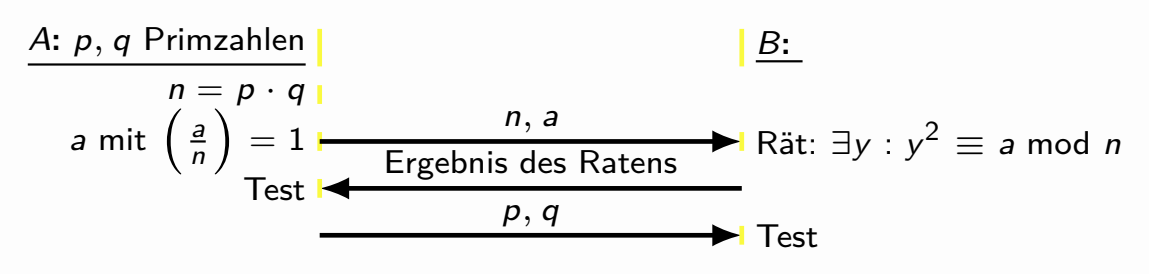
- Sei f eine One-Way-Funktion, die A und B kennen, aber f^{-1} ist nicht bekannt.



- Zweite Version eines 0-1-Protokolls
  - Sei $n = p \cdot q$ und $0 < a < n$ mit $\text{ggT}(a, n) = 1$.
  - Für die Hälfte aller a gilt $\left(\frac{a}{n}\right) = 1$
  - und für 50 Prozent dieser a's gilt: a ist quadratischer Rest modulo n.



Sicherheitsaspekte

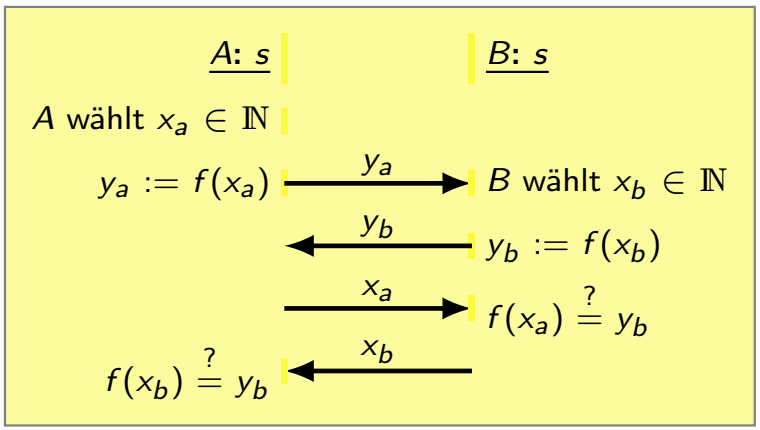


- Es ist sehr wichtig, dass B testet, ob p und q wirklich Primzahlen sind.
- Denn mit drei Primzahlen (beispielsweise $n = p_1 \cdot p_2 \cdot q$) kann das Ergebnis verfälscht werden.
- Dann gilt: $\left(\frac{a}{p_1}\right) = \left(\frac{a}{p_2}\right) = -1$ und $\left(\frac{a}{q}\right) = 1$.
- In Abhängigkeit der geratenen Antwort von B sendet A dann

$$\begin{array}{rrcl} & p' = p_1 \cdot p_2 & \text{und} & q' = q \\ \text{oder} & p' = p_1 & \text{und} & q' = q \cdot p_2 \end{array}$$

Würfeln großer Zufallszahlen

- Dieses Protokoll dient zum gemeinsamen Würfeln von Zufallszahlen.
- Die Zufallszahl x setzt sich dann wie folgt zusammen: $x = x_a + x_b \bmod s$, wobei s die Bereichsgrenze (Würfelintervall) angibt.



Einleitung

- Jeder Spieler bekommt 5 aus 52 Karten zugeteilt.
- Falls p, q Primzahlen sind mit $p \equiv q \equiv 3 \bmod 4$ und $n = p \cdot q$, a, b verschiedene Quadratwurzeln derselben Zahl ($a \not\equiv \pm b \bmod n$), dann gilt:

$$\left(\frac{a}{n}\right) = -\left(\frac{b}{n}\right)$$

- Beachte: Mit zwei verschiedenen Quadratwurzeln derselben Zahl kann n faktorisiert werden!
- Wir betrachten nur das Verteilen der ersten fünf Karten.

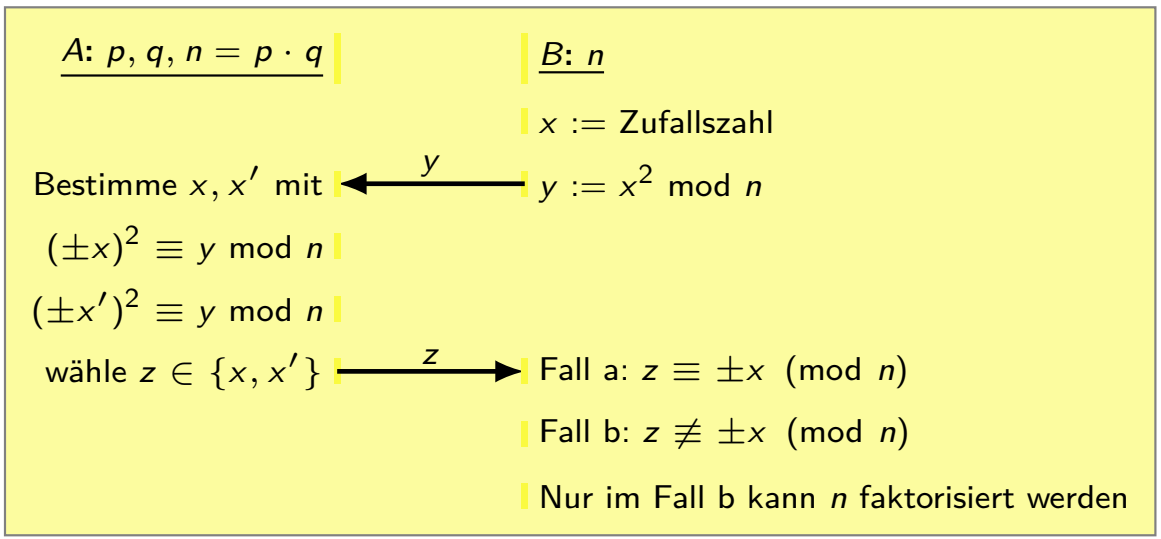
- A würfelt an B 5 Karten.
- Nur B kennt seine 5 Karten.
- Aus den verbleibenden 47 Karten würfelt B an A 5 Karten.
- Nur A kennt seine 5 Karten.
- A und B machen eine Spielrunde.
- A und B legen die gebeimen Daten der Protokolle offen.
- A und B testen die Protokolle.

Protokoll

A:		B:
Mischt die Karten K_i $(1 \leqslant i \leqslant 52)$		
Für $1 \leqslant i \leqslant 52$ bestimme Primzahlen p_i, q_i mit		
$p_i \equiv q_i \equiv 3 \bmod 4$ und setzt $n_i = p_i \cdot q_i$		
$T_i = (t_1^i t_2^i \dots t_6^i)$ mit $\left(\frac{t_j^i}{n_i}\right) = 1$ und		
$t_j^i \in QR_{n_i} \iff$ Bit j in $bin(K_i)$ ist 1	$n_i, 1 \leqslant i \leqslant 52, t_j^i, 1 \leqslant j \leqslant 6$ →	
	Würfeln x_i →	nur B kennt die x_i
	$1 \leqslant i \leqslant 52$	kann x_i nicht wählen
		wählt $I \subset \{1, \dots, 52\}$
	← $x_i^2 \bmod n_i, \left(\frac{x_i}{n_i}\right)$ für $i \notin I$	mit $\lvert I\rvert = 5$
	← $x_i^2 \bmod n_i, -\left(\frac{x_i}{n_i}\right)$ für $i \in I$	
Bestimmt $x_i' : x_i'^2 \equiv x_i^2 \pmod{n_i}$	$x_i', \left(\frac{x_i'}{n_i}\right) = \left(\frac{x_i}{n_i}\right)$ für $i \notin I$ →	für $j \in I$: faktorisiert n_j
	$x_i', \left(\frac{x_i'}{n_i}\right) = -\left(\frac{x_i}{n_i}\right)$ für $i \in I$	und $t_l^j \overset{?}{\in} QR_{n_j}$ und K_j

Einleitung

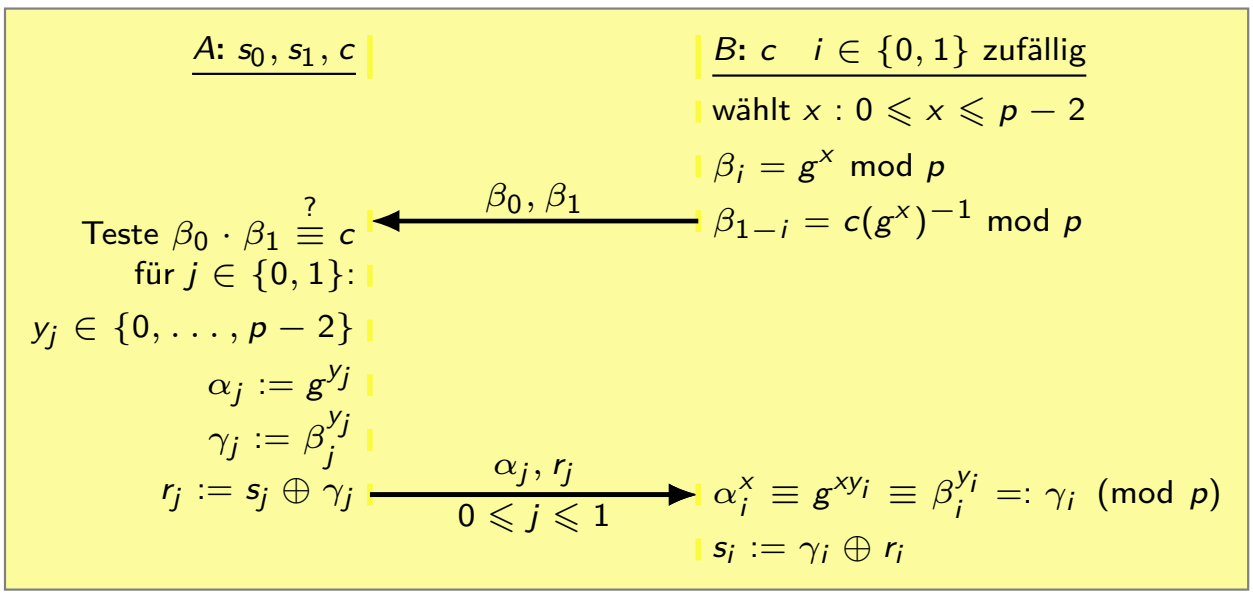
- Es gibt zwei Möglichkeiten für eine vergessliche Übertragung (engl. oblivious transfer):
  1. A sendet an B, Übertragung erfolgreich mit Wahrscheinlichkeit $\frac{1}{2}$, A kennt Resultat nicht
  2. A sendet eine von k Nachrichten an B, weiß aber nicht, welche B ausgewählt hat (Kaufen eines Geheimnisses).
- Erste Variante:



- Mit n kann dann ein beliebiges Geheimnis verschlüsselt sein.

Protokoll zum Kaufen eines von zwei Geheimnisses

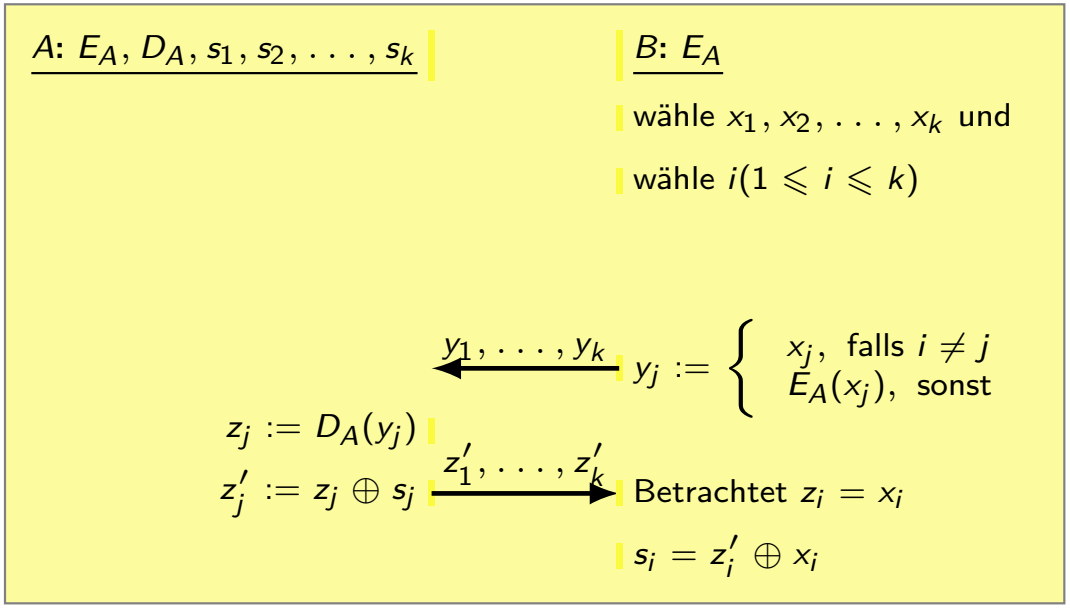
- Dieses Protokoll benutzt das Bitweise-XOR $\oplus$.
- Allen Parteien bekannt ist die Primzahl p, g als Generator von $F^*(p)$
- und eine Zahl c.
- Keiner kennt jedoch den diskreten Logarithmus von c.



- *Bemerkung:* Es ist für B nicht möglich, beide Geheimnisse zu erfahren.

Protokoll zum Kaufen eines von k Geheimnisses

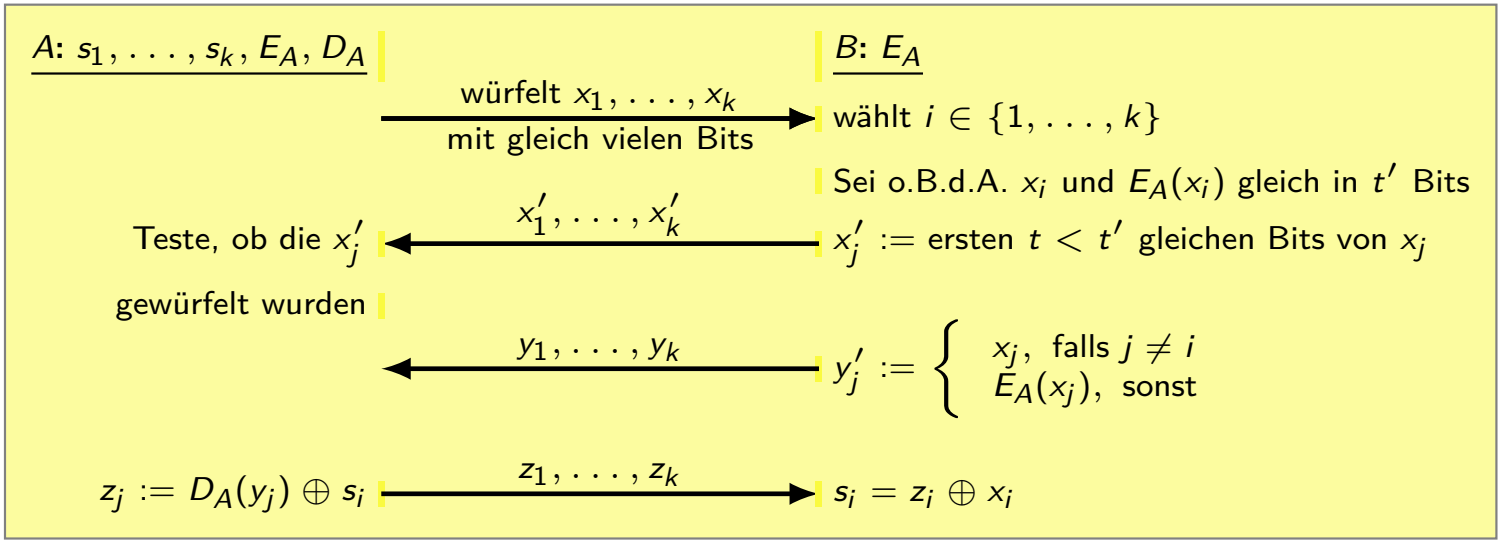
- Idee "markiere" das gewünschte Objekt durch eine Verschlüsselung.



- Beachte, dass B das Protokoll aktiv ändern kann und u.U. mehr als ein Geheimnis erhält.
- Somit muss A dem B vollständig trauen.

Protokoll zum Kaufen eines von k Geheimnisses

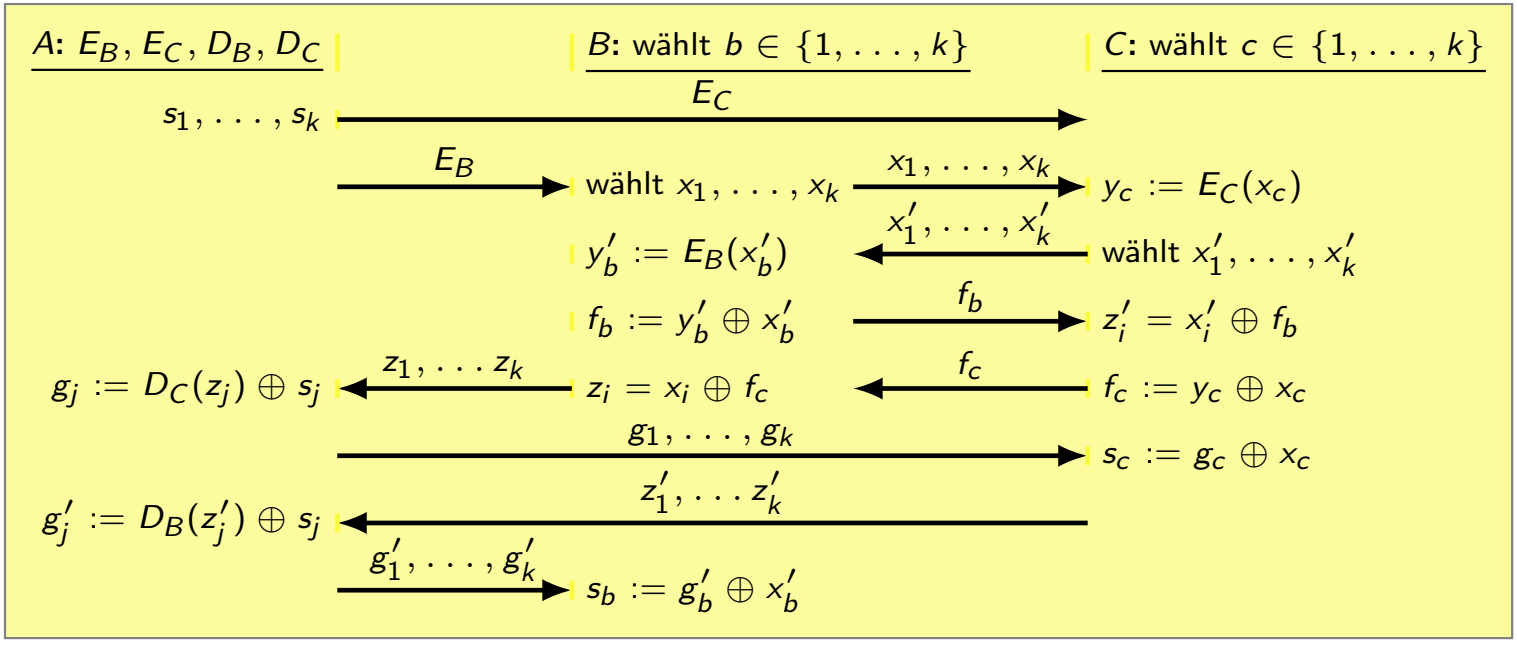
- Idee: A testet die Aktionen von B.



- Beachte: A kann Information über die Anfrage von B erhalten, indem er $D_A(y_j)$ bestimmt.

Gegenseitiges Überwachen

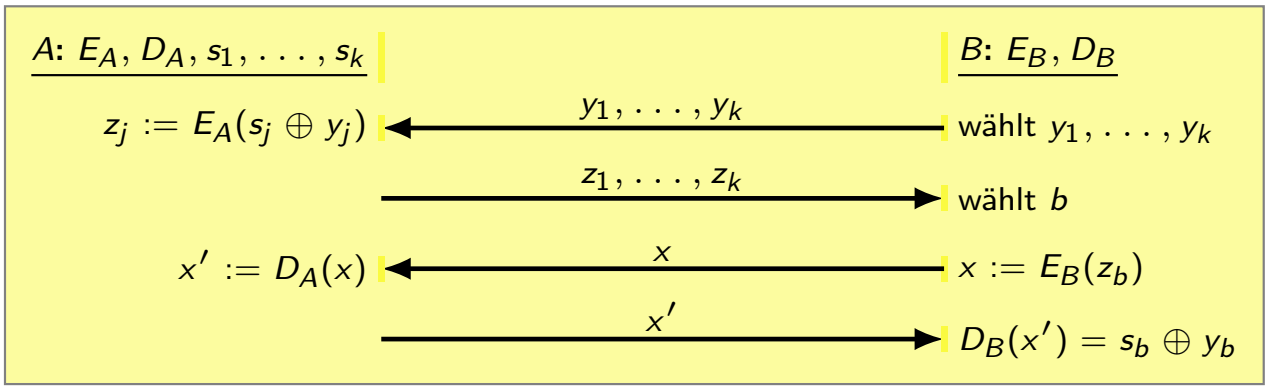
- B und C überwachen sich gegenseitig (Einbringen von Konkurrenten).



- Hier wurden jeweils die Fixpunkte von y'_b bzw. y_c bestimmt, also die Bits, die unverändert bleiben.
- Anhand von diesen Fixpunkten wurden die Funktionen $E_B(x'_b)$ und $E_B(x_c)$ durch den Partner nachvollzogen.

Klassisches System

- Betrachte folgendes System:



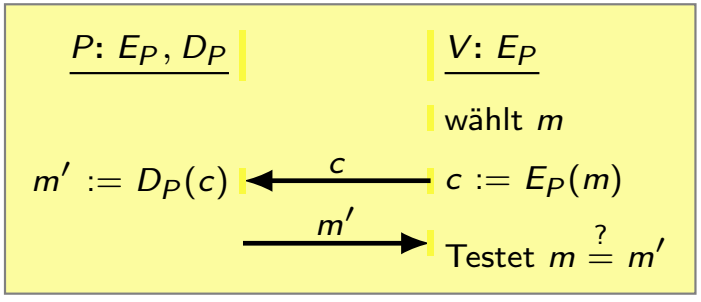
- Beachte: $D_B(x') = D_B(D_A(x)) = D_B(D_A(E_B(z_b))) = D_A(z_b) = s_b \oplus y_b$
- B kann versuchen, mehr Informationen zu erhalten, indem er nicht $E_B(z_b)$ schickt, sondern eine Kombination verschiedener Teile der z_i.
- Eine gute Wahl von E_A verhindert dieses allerdings.

Einleitung

- Bei der Identifikation beweist ein Teilnehmer seine Identität.
- Dieses geschieht im einfachsten Fall durch eine PIN oder ein Passwort.
- Allgemein gibt es einen Prover (Beweisführer) und einen Verifier (Beweistester).
- Der Prover kennt ein persönliches Geheimnis und zeigt dem Verifier die Kenntnis des Geheimnisses an, ohne dieses selber preiszugeben. Das Ziel ist:
  1. Falls der Prover P das Geheimnis kennt, dann wird der Verifier V immer überzeugt von der Identität von P.
  2. Falls P mit beliebig hinreichender Sicherheit V von seiner Identität überzeugen kann, dann kennt P auch das Geheimnis.

Beispiel mit Public-Key

- Einfaches Beispiel:



- Hierbei handelt es sich um ein 2-Runden Protokoll.
- Es dürfen aber nicht E_P, D_P zum Verschlüsseln verwendet werden, denn sonst wird das Verfahren der Identifikation unsicher.

Verbesserung

- Benutzer werden durch eine Zahl i identifiziert. Diese kann z.B. der IP-Nummer, dem Namen der Person o.ä. zugeordnet sein.
- Ein vertrauenswürdiger Agent S bestimmt p, q, n, e, d, wie beim RSA-Verfahren und eine One-Way-Funktion f mit zwei Parametern.
- Jeder Benutzer bekommt von S eine Zahl x_i mit $x_i \equiv i^d \pmod{n}$.
- Alle Benutzer kennen n, e, f.
- Die Unterschrift von Benutzer i zur Nachricht w ist dann das Zahlenpaar (s, t) mit $s^e \equiv it^{f(t,w)} \pmod{n}$.
- **Leisten einer Unterschrift**
  - $r :=$ Zufallszahl
  - $t := r^e \bmod n$
  - $s := x_i r^{f(t,w)} \bmod n$
- $\Longrightarrow s^e \equiv x_i^e r^{ef(t,w)} \equiv it^{f(t,w)} \bmod n$
- **Verifikation der Unterschrift** Teste, ob $s^e \stackrel{?}{\equiv} it^{f(t,w)} \bmod n$.
- Beachte: f ist eine kryptographische Hashfunktion.
- Unterschriftsfälschung entspricht der Lösung des diskreten Logarithmus!

Verbindliche einfache Unterschrift

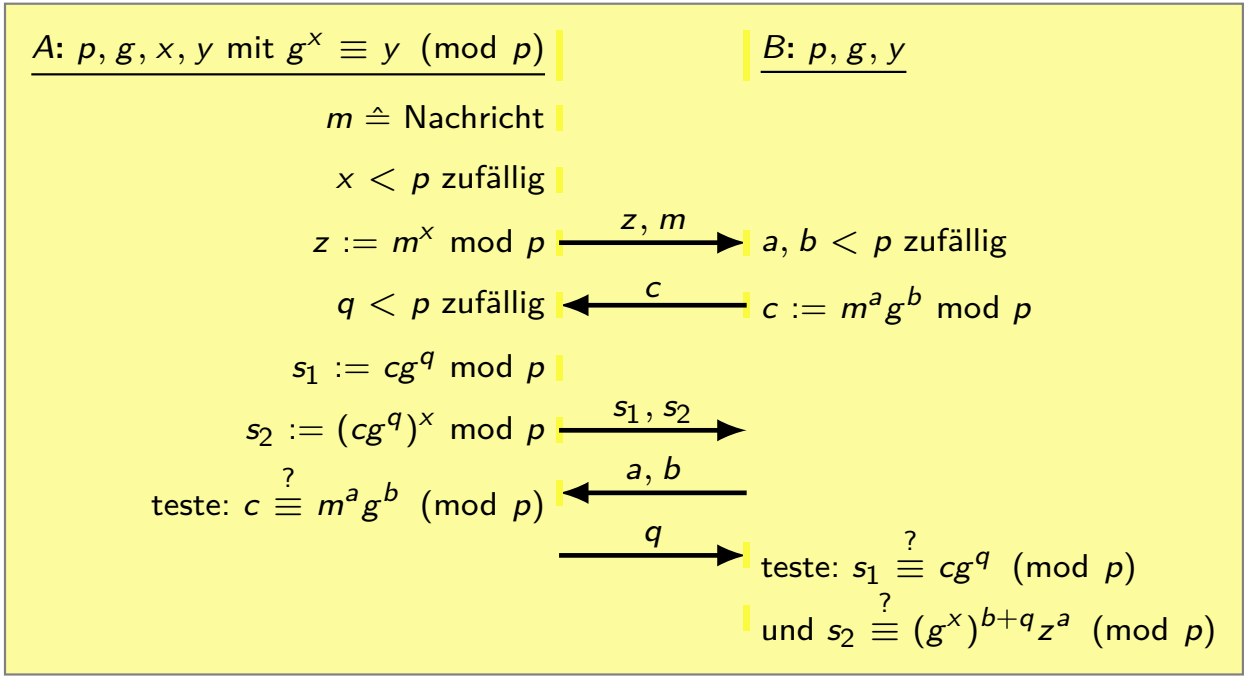
- Empfänger ist von Unterschrift überzeugt, aber
- Empfänger kann diese ohne Hilfe vom Sender nicht weitergeben.

A: p Primzahl, g Generator		B: p, g, y
x privater Schlüssel		
$y = g^x \bmod p$		
$m \mathrel{\hat{=}}$ Nachricht		
$z := m^x \bmod p$	$\xrightarrow{z,\ m}$	wähle $a, b < p$
$t := x^{-1} \bmod (p-1)$	$\xleftarrow{c}$	$c := z^a y^b \bmod p$
$d := c^t \bmod p$	$\xrightarrow{d}$	Teste, ob $d \equiv m^a g^b \pmod p$ gilt
		falls ja: Unterschrift ok

- *Ergebnis:* B kann betrügen, indem er zuerst c berechnet, dann d bestimmt, und damit die Übertragung simuliert.

Verbesserte verbindliche Unterschrift

- B kann von der Unterschrift überzeugt werden.
- A kann zeigen, dass eine Unterschrift gefälscht ist.



- Beachte:

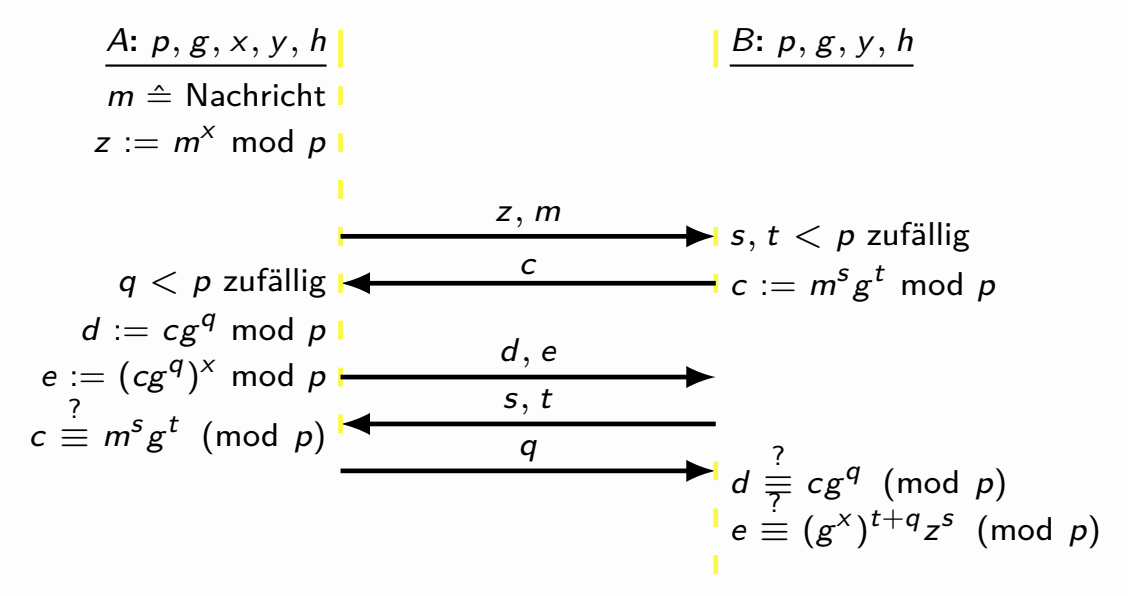
$$\begin{aligned} s_2 &\equiv (cg^q)^x \equiv (m^a g^b g^q)^x \equiv (m^x)^a (g^x)^{(b+q)} \\ &\equiv m^{xa} g^{(b+q)x} \equiv z^a y^{(b+q)} \equiv (g^x)^{b+q} z^a \pmod{p} \end{aligned}$$

Verifikation durch Dritte

- A unterschreibt für B,
- B verifiziert und
- lässt die Unterschrift zukünftig durch einen von A ausgewählten C bestätigen.
- p und g seien öffentlich, $n := p \cdot q'$ mit p, q' Primzahlen.
- H ist eine Hashfunktion, die bei gut gewähltem n auch weggelassen werden kann.
- h ist öffentlicher Schlüssel von C.

Verbindliche Unterschrift (Rückblick)

p, g, h
$n = p \cdot q'$

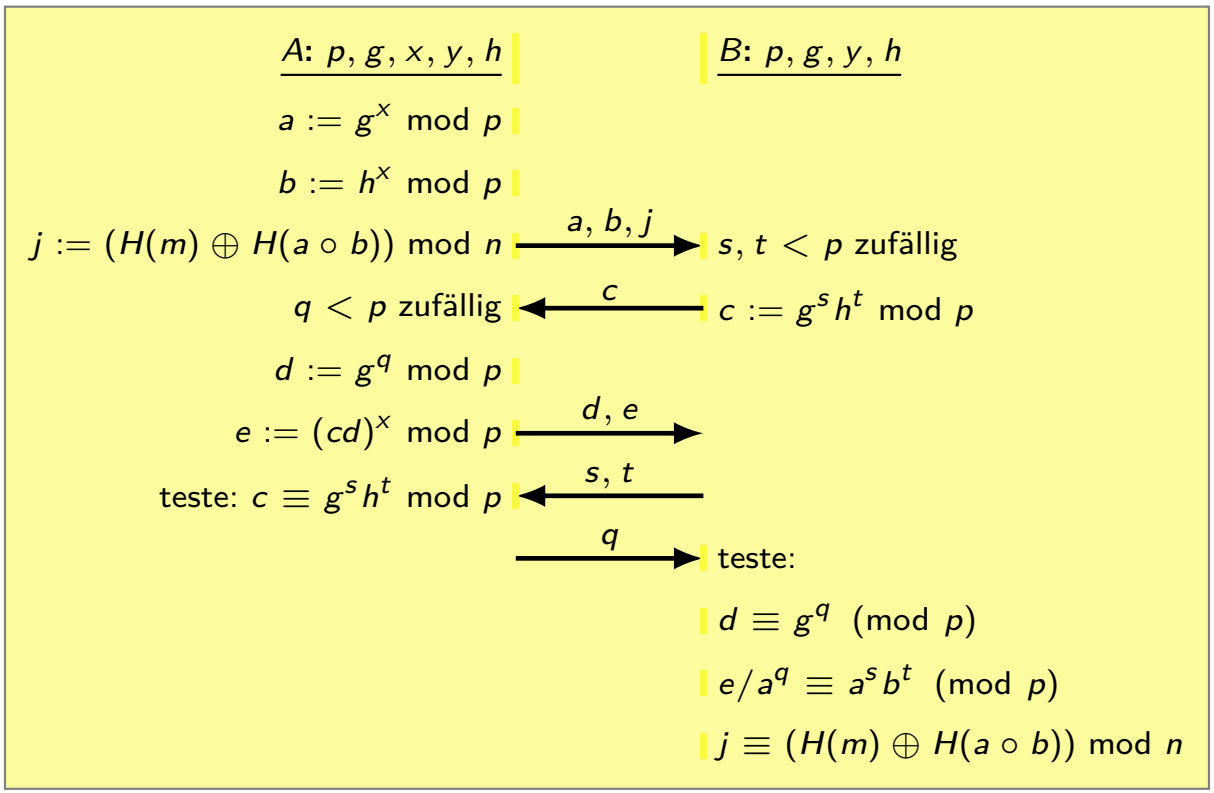


Beachte $e := (cg^q)^x \equiv (m^s g^t g^q)^x \equiv (m^x)^s (g^x)^{(t+q)} \equiv m^{xs} g^{(t+q)x} \equiv z^s y^{(t+q)} \equiv (g^x)^{t+q} z^s \pmod{p} d := cg^q \bmod p \implies d := g^q \bmod p z :=$
$m^x \bmod p \implies a := g^x \bmod p, j := (H(m) \oplus H(a)) \bmod nc :=$
$m^s g^t \bmod p \implies c := g^s h^t \bmod p \implies b := h^x \bmod p e \stackrel{?}{\equiv} (g^x)^{t+q} z^s \pmod{p} \implies e/a^q \equiv a^s b^t \pmod{p}$Beachte:
$e/a^q \equiv (cd)^x / a^q \equiv (g^s h^t g^q)^x / (g^x)^q (g^s h^t)^x \equiv a^s b^t \pmod{p}.(H(m) \oplus H(a)) \bmod n \implies H(m) \oplus H(a \circ b)$

Verifikation durch Dritte

h öffentlicher Schlüssel von C
p, g
$n = p \cdot q'$

- Leisten der Unterschrift:



- Beachte: $e/a^q \equiv (cd)^x/a^q \equiv (g^s h^t g^q)^x/(g^x)^q \equiv (g^s h^t)^x \equiv a^s b^t \pmod{p}$.

Verifizieren der Unterschrift durch C gegenüber D:

p, g
$n = p \cdot q'$
$a := g^x \bmod p$
$b := h^x \bmod p$

- Verifikation:

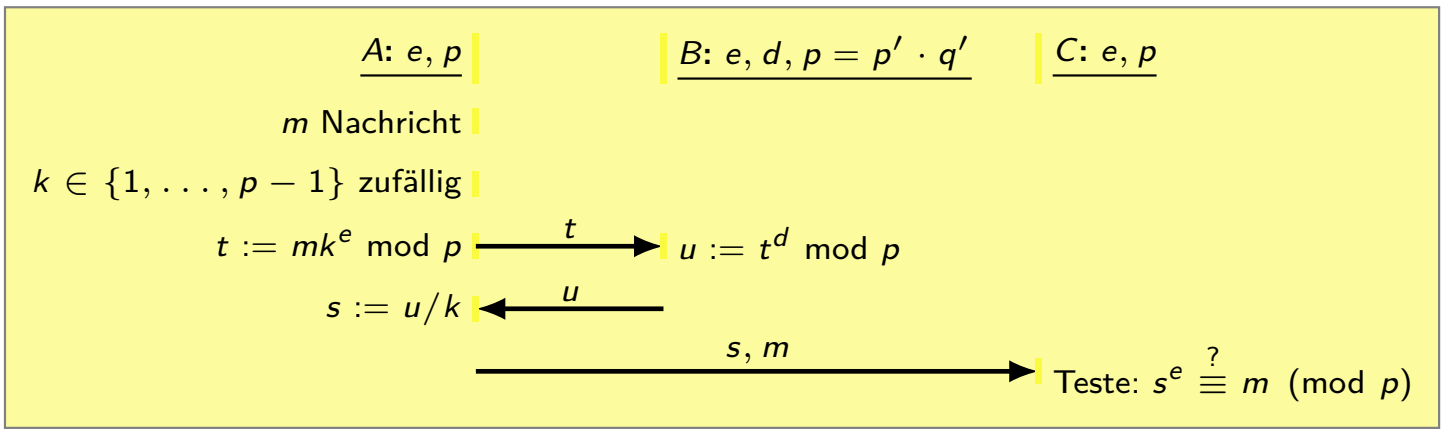
C: z (privater Schlüssel)		D: a, b
$u, v < p$ zufällig		
$h = g^z \bmod p$		
$w < p$ zufällig		
$l := g^w \bmod p$	$\xleftarrow{k}$	$k := g^u a^v \bmod p$
$y := (kl)^z \bmod p$	$\xrightarrow{l, y}$	
teste: $k \equiv g^u a^v \pmod p$	$\xleftarrow{u, v}$	
	$\xrightarrow{w}$	teste: $g^w \equiv l \pmod p$
		teste: $y/h^w \equiv h^u b^v \pmod p$

- Beachte:

$$\begin{aligned} y/h^w &\equiv (kl)^z/(g^z)^w \equiv (g^u a^v g^w)^z/(g^z)^w \equiv (g^u a^v)^z \equiv h^u (a^v)^z \\ &\equiv h^u((g^x)^v)^z \equiv h^u((b^{1/z})^v)^z \equiv h^u b^v \pmod p \end{aligned}$$

Blinde Unterschrift

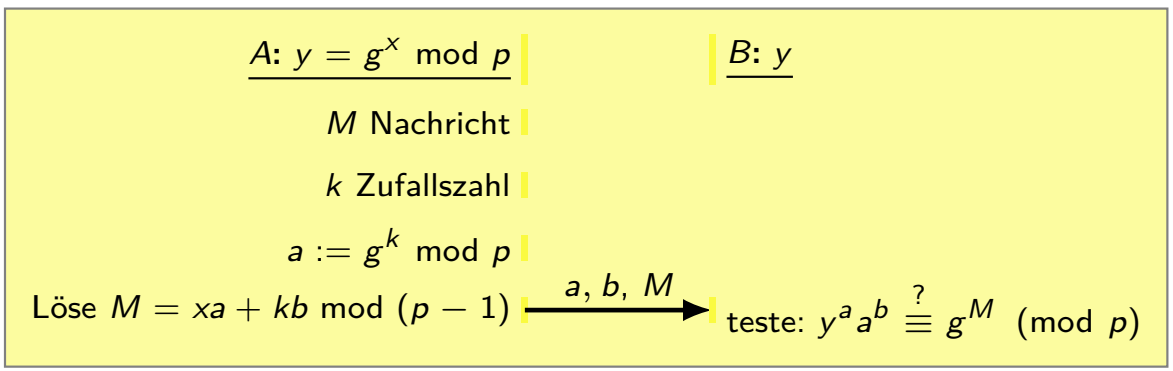
- Die blinde Unterschrift ist eine weitere Variante von Unterschriftsprotokollen.
- Hierbei ist dem Unterzeichner der Inhalt des Dokuments, das er unterschreibt, nicht bekannt.



- Beachte: $s \equiv u/k \equiv t^d/k \equiv (mk^e)^d/k \equiv m^d k^{ed}/k \equiv m^d$.

Einleitung

- Das Unterschriftsprotokoll kann auch zur Nachrichtenübermittlung „missbraucht" werden.
- Wir wollen dieses am Beispiel von ElGamal zeigen.
- Zur Erinnerung zunächst das gewöhnliche Verfahren:



- Beachte: $y^a a^b \equiv g^{x\,a} g^{k\,b} \equiv g^{xa} g^{kb} \equiv g^{xa+kb} \equiv g^M \pmod p$
- **Idee:** Übermittlung geheimer Information durch Zufallszahl k.

Übermittlung geheimer Nachrichten

- Erinnerung:

A: $y = g^x \bmod p$		B: y
M Nachricht		
k Zufallszahl		
$a := g^k \bmod p$		
Löse $M = xa + kb \bmod (p-1)$	$\xrightarrow{a,\ b,\ M}$	teste: $y^a a^b \stackrel{?}{\equiv} g^M \pmod p$

- Sei nun M eine zu übermittelnde geheime Nachricht:

A: $k = g^r \bmod p$		B: k, g, p
M, M', p teilerfrei		r (privater Schlüssel für M)
$M', p-1$ teilerfrei		
$X = g^M \bmod p$		
Löse $M' = rX + My \bmod (p-1)$	$\xrightarrow{X,\ y,\ M'}$	teste: $(g^r)^X X^y \equiv g^{M'}$
		teste: $M' \equiv k^X X^y \bmod p$
		$M = y^{-1}(M' - rX) \bmod (p-1)$

- Beachte: Der (evtl. mithörende) Staat kennt nur X, y, M', p, g.

Broadcasting

- Ziel
  - Man möchte an viele Teilnehmer verschlüsselt die gleiche Nachricht schicken.
  - Für jede Teilmenge S der Teilnehmer soll es möglich sein, die Nachricht so zu verschlüsseln, dass sie nur von Teilnehmern aus S entschlüsselt werden kann.
  - Teilnehmer, die nicht aus S sind, dürfen die Nachricht nicht entschlüsseln können (nicht mal, wenn sie kooperieren).
- Anwendungen:
  - Pay-TV
  - Verschlüsselte Dateisysteme
  - DVDs

Naiver Ansatz

- n potentielle Teilnehmer, $W_n = \{1, \ldots, n\}$
- $S \subseteq W_n$ = berechtigte Teilnehmer, M = Nachricht
- Naiver Ansatz:
  - Jeder Teilnehmer $i \in S$ erzeugt Schlüsselpaar (E_i, D_i) und veröffentlicht E_i.
  - Sender erzeugt zufälligen Schlüssel K für ein symmetrisches Verschlüsselungsverfahren.
  - BROADCAST: $\forall i \in S : E_i(K)$ und $E_K(M)$.
- Zusätzliche Information hat Größenordnung $\Theta(|S|)$.

Bilineare Abbildungen

Bilineare Abbildung

Seien G und G' zyklische Gruppen der Ordnung p für eine Primzahl p. Sei g ein Generator von G. Eine Abbildung $e\colon G \times G \to G'$ nennen wir im Folgenden bilinear, wenn gilt:

1. Bilinear: $\forall u, v \in G, \forall a, b \in \mathbb{Z} : e(u^a, v^b) = e(u, v)^{ab}$.
2. Nicht degeneriert: $e(g, g) \neq 1$.
3. Effizient: e muss effizient berechenbar sein.

Eigenschaften:

- symmetrisch: $e(u, v) = e(v, u)$.
- distributiv: $e(u, vv') = e(u, v) \cdot e(u, v')$.
- $e(u^a, v) = e(u, v^a)$.

Aufbau des Systems

berechtigte Teilnehmer: $S \subseteq W_n$

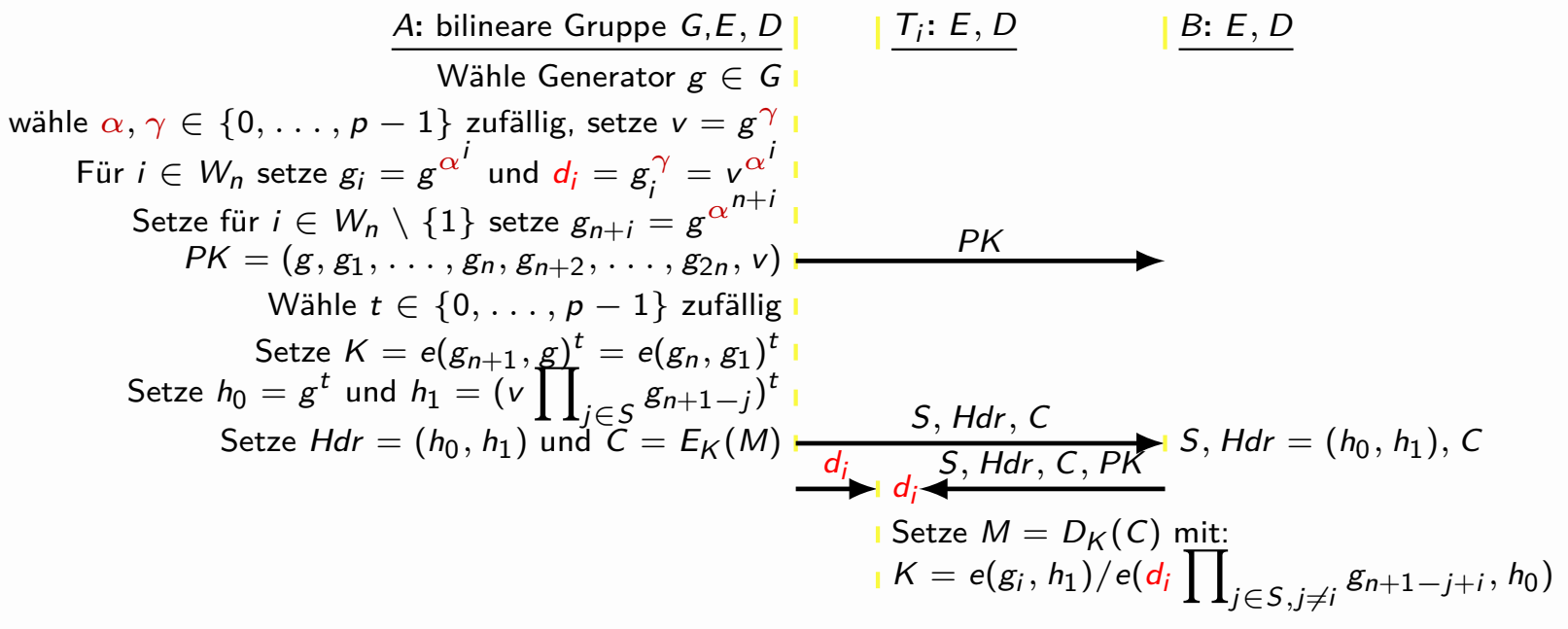
- Zum Aufbau sind zu erzeugen:
  - öffentlicher Schlüssel PK und
  - private Schlüssel $d_1, \ldots, d_n$.
- Zum Verschlüsseln wird erzeugt:
  - Header: Hdr und
  - symmetrischer Schlüssel K.
  - Gesendet wird: Hdr und Nachricht verschlüsselt mit K
- Für $i \in S$ wird zum Entschlüsseln:
  - aus S, d_i, Hdr, PK
  - der Schlüssel K bestimmt.

Einfache Variante

berechtigte Teilnehmer: $S \subseteq W_n$

A: bilineare Gruppe G, E, D	T_i: E, D	B: E, D
Wähle Generator $g \in G$		
wähle $\alpha, \gamma \in \{0, \ldots, p-1\}$ zufällig, setze $v = g^{\gamma}$		
Für $i \in W_n$ setze $g_i = g^{\alpha^i}$ und $d_i = g_i^{\gamma} = v^{\alpha^i}$		
Setze für $i \in W_n \setminus \{1\}$ setze $g_{n+i} = g^{\alpha^{n+i}}$		
$PK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v)$	$\xrightarrow{PK}$	
Wähle $t \in \{0, \ldots, p-1\}$ zufällig		
Setze $K = e(g_{n+1}, g)^t = e(g_n, g_1)^t$		
Setze $h_0 = g^t$ und $h_1 = (v \prod_{j \in S} g_{n+1-j})^t$		
Setze $Hdr = (h_0, h_1)$ und $C = E_K(M)$	$\xrightarrow{S, Hdr, C}$	$S, Hdr = (h_0, h_1), C$
	$d_i \rightarrow d_i \leftarrow S, Hdr, C, PK$	
	Setze $M = D_K(C)$ mit:	
	$K = e(g_i, h_1)/e(d_i \prod_{j \in S, j \neq i} g_{n+1-j+i}, h_0)$	

Korrektheit der Entschlüsselung



- Verschlüsselt wurde mit: $K = e(g_{n+1}, g)^t$.
- Entschlüsselt mit: $K = e(g_i, h_1)/e(d_i \prod_{j\in S, j\neq i} g_{n+1-j+i}, h_0)$.
- Weiter haben wir:
  - $g_i = g^{\alpha^i}$, $d_i = g_i^{\gamma} = v^{\alpha^i}$ und $v = g^{\gamma}$, $g_i^{(\alpha^j)} = g_{i+j}$,
  - sowie $h_0 = g^t$ und $h_1 = (v \prod_{j\in S} g_{n+1-j})^t$.

Korrektheit der Entschlüsselung

$g_i = g^{\alpha^i}$, $d_i = g_i^{\gamma} = v^{\alpha^i}$ und $v = g^{\gamma}$, $g_i^{(\alpha^j)} = g_{i+j}$, $h_0 = g^t$ und $h_1 = (v \prod_{j \in S} g_{n+1-j})^t$.

- Verschlüsselt wurde mit: $e(g_{n+1}, g)^t$.
- Entschlüsselt mit: $K = e(g_i, h_1)/e(d_i \prod_{j \in S, j \neq i} g_{n+1-j+i}, h_0)$.

$$
\begin{aligned}
& e(g_i, h_1)/e(d_i \textstyle\prod_{j \in S, j \neq i} g_{n+1-j+i}, h_0) \\
=& e(g^{\alpha^i}, (v \textstyle\prod_{j \in S} g_{n+1-j})^t)/e(v^{\alpha^i} \textstyle\prod_{j \in S, j \neq i} g_{n+1-j+i}, g^t) \\
=& e(g^{\alpha^i}, (v \cdot g_{n+1-i}^t \textstyle\prod_{j \in S, j \neq i} g_{n+1-j})^t)/e(v^{\alpha^i} \textstyle\prod_{j \in S, j \neq i} g_{n+1-j+i}, g^t) \\
=& e(g^{\alpha^i}, g_{n+1-i}^t) \cdot e(g^{\alpha^i}, (v \textstyle\prod_{j \in S, j \neq i} g_{n+1-j})^t)/e(v^{\alpha^i} \textstyle\prod_{j \in S, j \neq i} g_{n+1-j+i}, g^t) \\
=& e(g^{\alpha^i}, g_{n+1-i})^t \cdot e(g^{\alpha^i}, v \textstyle\prod_{j \in S, j \neq i} g_{n+1-j})^t/e(v^{\alpha^i} \textstyle\prod_{j \in S, j \neq i} g_{n+1-j+i}, g)^t \\
=& e(g_{n+1}, g)^t \cdot e(g^{\alpha^i}, v \textstyle\prod_{j \in S, j \neq i} g_{n+1-j})^t/e(v^{\alpha^i} \textstyle\prod_{j \in S, j \neq i} g_{n+1-j+i}, g)^t \\
=& e(g_{n+1}, g)^t \cdot e(v^{\alpha^i}, \textstyle\prod_{j \in S, j \neq i} g_{n+1-j})^t/e(v^{\alpha^i} \textstyle\prod_{j \in S, j \neq i} g_{n+1-j+i}, g)^t \\
=& e(g_{n+1}, g)^t
\end{aligned}
$$

Effizienz und Eigenschaften des einfachen Protokolls

$g_i = g^{\alpha^i}$, $d_i = g_i^{\gamma} = v^{\alpha^i}$ und $v = g^{\gamma}$, $g_i^{(\alpha^j)} = g_{i+j}$, $h_0 = g^t$ und $h_1 = (v \prod_{j \in S} g_{n+1-j})^t$.

- Zeit zur Entschlüsselung dominiert durch Berechnung von

$$W_S^{(i)} = v^{\alpha^i} \cdot \prod_{j \in S, j \neq i} g_{n+1-j+i}.$$

  Dazu benötigt man $|S| - 2$ Gruppenoperationen.
- Ist $W_S^{(i)}$ bekannt, so kann $W_{S'}^{(i)}$ mit $|S \setminus S'| + |S' \setminus S|$ Gruppenoperationen berechnet werden.
- Der geheime Schlüssel d_i kann zusätzlich auch noch $W_{W_n}^{(i)}$ enthalten. Nützlich falls $|S| = n - r$ für kleines r.
- Eigenschaften:
  - Größe des Public Keys: $2n + 1$ Gruppenelemente.
  - Größe des Headers: 2 Gruppenelemente ($+ \Theta(|S| \log n)$ Bits für Kodierung von S).
  - Größe der privaten Schlüssel: ein Gruppenelement.
- Anwendung: verschlüsseltes Dateisystem:
  - Naiver Ansatz verschlüsselt K für jeden Benutzer einmal. Platzbedarf: $\approx 44|S|$ Bytes.
  - Einfaches Protokoll: $4|S|$ Bytes + 2 Gruppenelemente (bei 32-Bit langen Benutzer-IDs).

Verbessertes Protokoll

$g_i = g^{\alpha^i}$, $d_i = g_i^{\gamma} = v^{\alpha^i}$ und $v = g^{\gamma}$, $g_i^{(\alpha^j)} = g_{i+j}$, $h_0 = g^t$ und $h_1 = (v \prod_{j \in S} g_{n+1-j})^t$.

- Benutze $n_1 = \lceil \frac{n}{n_2} \rceil$ Instanzen für jeweils n_2 Teilnehmer und teile Informationen zwischen den Instanzen.
- Setze nun: $PK = (g, g_1, \ldots, g_{n_2}, g_{n_2+2}, \ldots, g_{2n_2}, v_1, \ldots, v_{n_1})$ mit:
  - Generator $g \in G$,
  - $\alpha, \gamma_1, \ldots, \gamma_{n_1} \in \{0, \ldots, p-1\}$ zufällig und
  - $g_i = g^{\alpha^i}$
  - $d_i = g_{i \bmod n_2}^{\gamma_{\lfloor i/n_2 \rfloor}} = v_{\lfloor i/n_2 \rfloor}^{\alpha^{i \bmod n_2}}$
- Private Schlüssel:

$$\begin{array}{ccccc} g_1^{\gamma_1} & g_2^{\gamma_1} & g_3^{\gamma_1} & g_4^{\gamma_1} & g_5^{\gamma_1} \\ g_1^{\gamma_2} & g_2^{\gamma_2} & g_3^{\gamma_2} & g_4^{\gamma_2} & g_5^{\gamma_2} \\ g_1^{\gamma_3} & g_2^{\gamma_3} & g_3^{\gamma_3} & g_4^{\gamma_3} & g_5^{\gamma_3} \\ g_1^{\gamma_4} & g_2^{\gamma_4} & g_3^{\gamma_4} & g_4^{\gamma_4} & g_5^{\gamma_4} \end{array}$$

- Verschlüsslung und Entschlüsselung funktionieren analog wie im einfachen Protokoll.

Eigenschaften des verbesserten Protokolls

- Welche Eigenschaften besitzt das verbesserte Protokoll für $n_1 = n_2 = \sqrt{n}$?
  - Größe des Public Keys: $2B + A = \Theta(\sqrt{n})$ Gruppenelemente.
  - Größe des Headers: $A + 1 = \Theta(\sqrt{n})$ Gruppenelemente (+ Platz für Kodierung von S).
  - Größe der privaten Schlüssel: ein Gruppenelement.
- Anwendung: DVDs:
  - Annahme: Jeder DVD-Spieler hat eigenen Schlüssel. Bestimmte DVD-Spieler sollen bei zukünftigen DVDs deaktiviert werden können. $n = 2^{32}$ verschiedene DVD-Spieler.
  - Benutze Protokoll mit $A = B = \sqrt{n}$. Speichere auf jeder DVD Hdr und PK und die Menge der gesperrten DVD-Spieler.
  - Platzbedarf: $\Theta(\sqrt{n})$ Gruppenelemente + $4r$ Bytes (r=# gesperrter DVD-Spieler).
  - Für realistische Parameter $\approx 5MB + 4r$ Bytes.

Verallgemeinertes Diffie-Hellman Problem

- G zyklische Gruppe der Ordnung p, g, h Generatoren von G
- $e\colon G \times G \to G'$ eine bilineare Abbildung
- $\alpha \in \{0, \ldots, p-1\}$ zufällig gewählt

l-BDHE

Gegeben:

$$(g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^l}, g^{\alpha^{l+2}}, \ldots, g^{\alpha^{2l}}, h) \in G^{2l+1}$$

Gesucht:

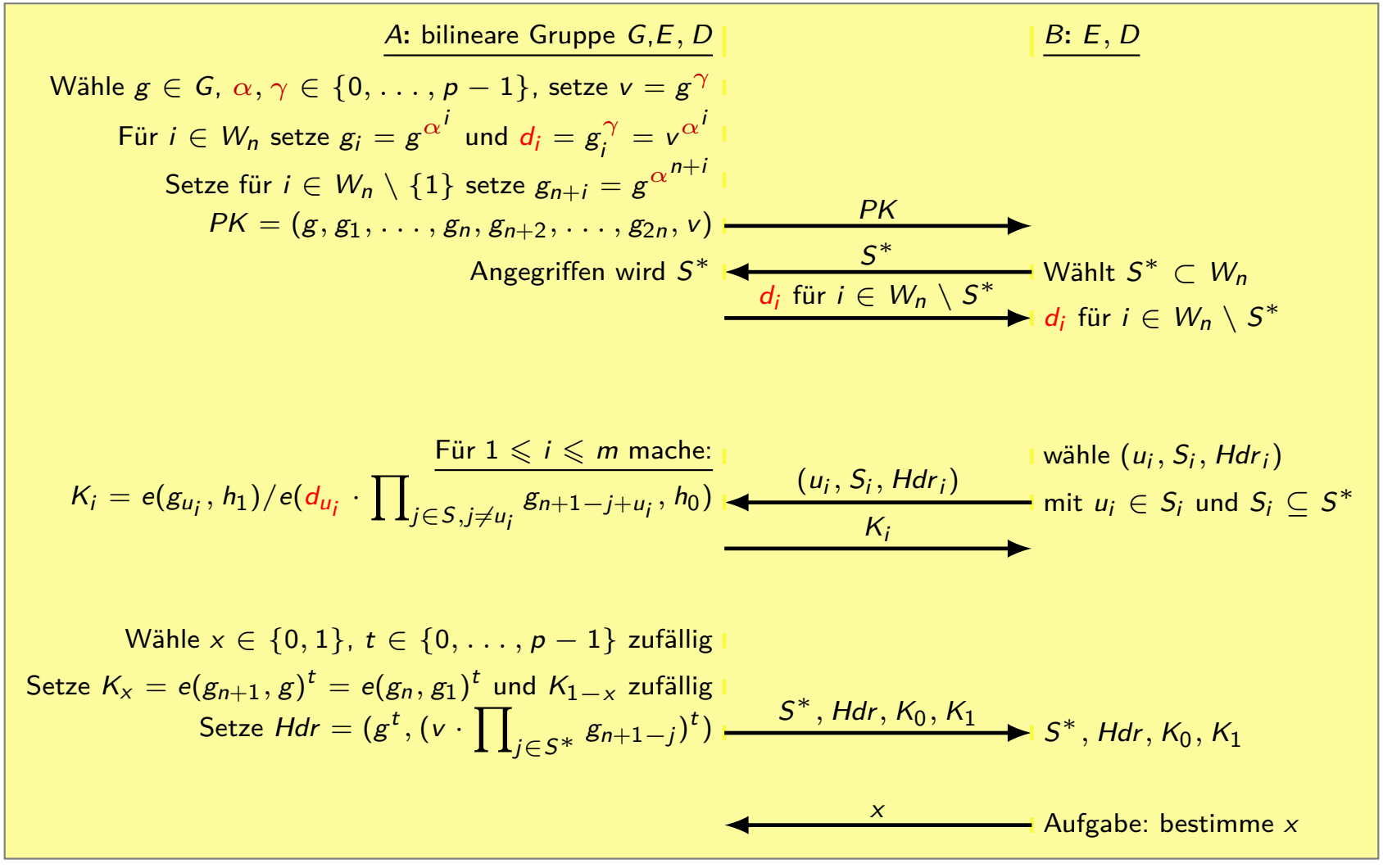
$$e(g,h)^{\alpha^{l+1}} = e(g^{\alpha^{l+1}}, h)$$

- Kein effizienter Algorithmus zur Lösung von l-BDHE bekannt.

Idee für Sicherheit (Schlimmster Angriff)

- Wann ist ein System sicher?
  - Wenn es einen Angriff abwehren kann.
  - Je stärker die abgewehrbaren Angriffe sind, desto sicherer ist ein System.
- Daher konstruieren wir einen sehr starken Angriff:
  - Angreifer muss sehr einfache Anfrage beantworten.
  - Angreifer darf vor der Anfrage möglichst viel bestimmen.
  - Angreifer darf vor der Anfrage möglichst viel anfragen (abfangen).
  - Angreifer bekommt Crypttext (Hdr) und zwei mögliche Inhalte (K_1, K_2).
  - Angreifer muss bestimmen, was der Inhalt ist.
  - D.h. muss bestimmen x, mit K_x ist in Hdr verschlüsselt.
  - D.h. wir bekommen auch automatisch Bit-Sicherheit.

Schlimmster Angriff



Angreifer-Modell

Modell

Init: $\mathcal{A}$ (Angreifer) wählt $S^* \subseteq W_n$, Menge der Angreifer ohne Schlüssel.

Setup: $\mathcal{A}$ erhält Public Key PK und private Schlüssel d_i für $i \notin S^*$.

Query: $\mathcal{A}$ erhält Entschlüsselung von m Anfragen (u_i, S_i, Hdr_i).

Challenge: $\mathcal{C}$ (Challenger) verschlüsselt und erhält (Hdr, K). $\mathcal{C}$ wählt $b \in \{0,1\}$ zufällig und setzt $K_b = K$ und wählt K_{1-b} zufällig. $\mathcal{C}$ schickt (Hdr, K_0, K_1) an $\mathcal{A}$.

Guess: $\mathcal{A}$ wählt $b' \in \{0,1\}$. Er gewinnt, falls $b = b'$ gilt.

Definition

Das Broadcasting-System ist (t, ε, n)-sicher, wenn für jeden Algorithmus $\mathcal{A}$, der höchstens t Rechenschritte läuft, gilt $\mathbf{Pr}[b = b'] < \frac{1}{2} + \varepsilon$.

Anschauung: $\mathcal{A}$ erkennt den richtigen Schlüssel nicht, selbst wenn er ihn sieht.

Verallgemeinertes Diffie-Hellman Problem

n-BDHE (Entscheidungsvariante)

Gegeben: $(g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^n}, g^{\alpha^{n+2}}, \ldots, g^{\alpha^{2n}}, h) \in G^{2n+1}$
Für zufälliges $b \in \{0,1\}$ setze
$K_b = e(g^{\alpha^{n+1}}, h)$ und $K_{1-b} \in G'$ zufällig.

Gesucht: b.

Theorem

Es gibt ein Polynom p, so dass für alle t, ε und n gilt: Wenn das n-BDHE (t, ε, n)-sicher ist, dann ist das Broadcasting-System $(p(n)t, \varepsilon, n)$-sicher.

Beweis durch Widerspruch: Mit Hilfe eines Algorithmus zum Angreifen des Protokolls kann ein Algorithmus zur Lösung des n-BDHE konstruiert werden.

Beweis der Sicherheit

- n-BDHE (Gesucht wird b):

  Gegeben: $(g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^n}, g^{\alpha^{n+2}}, \ldots, g^{\alpha^{2n}}, h) \in G^{2n+1}$

  $K_b = e(g^{\alpha^{n+1}}, h)$ und $K_{1-b} \in G'$ zufällig (b zuf.)

- Baue daraus ein Broadcasting-Protokoll.
  - $PK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v = g^{\gamma}), d_i = g_i^{\gamma} = v^{\alpha^i}$.
  - $PK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v)$ mit $v = g^u(\prod_{j \in S} g_{n+1-j})^{-1}$

    $\forall i \notin S : d_i = v^{\alpha^i} = g_i^u \prod_{j \in S} g_{n+1-j+i}^{-1}$ (wichtig $i \notin S$).
  - Beachte: u uniform zuf., damit auch v
- Challenge
  - $K = e(g_{n+1}, g)^t = e(g_n, g_1)^t$, $t \in \{0, \ldots, p-1\}$ zuf.
    $Hdr = (g^t, (v \cdot \prod_{j \in S} g_{n+1-j})^t)$.
  - Jetzt: $Hdr = (h, h^u)$, $b \in \{0, 1\}$ entsprechend der Eingabe für n-BDHE, $K_b = e(g_{n+1}, h)$, $K_{1-b} =$ zufälliges Element aus G'.
- Gültige Challenge, weil:
  - $h^u = (g^u)^t = (v \cdot \prod_{j \in S} g_{n+1-j})^t$
  - $K = e(g_{n+1}, g)^t = e(g_{n+1}, h) = K_b$

Guess

Guess

- Das Protokoll ist so konstruiert, dass die richtige Antwort b auf die Challenge der richtigen Antwort für das n-BDHE-Problem entspricht.
- Also: Algorithmus, mit Wahrscheinlichkeit $\geq 1/2 + \varepsilon$ das richtige b findet, dann löst er mit der gleichen Wahrscheinlichkeit auch das n-BDHE-Problem.
- $\Longrightarrow$ Ist n-BDHE (t, ε, n)-sicher, so ist das Protokoll $(p(n)t, \varepsilon, n)$-sicher.

Erweiterungen / Offene Fragen

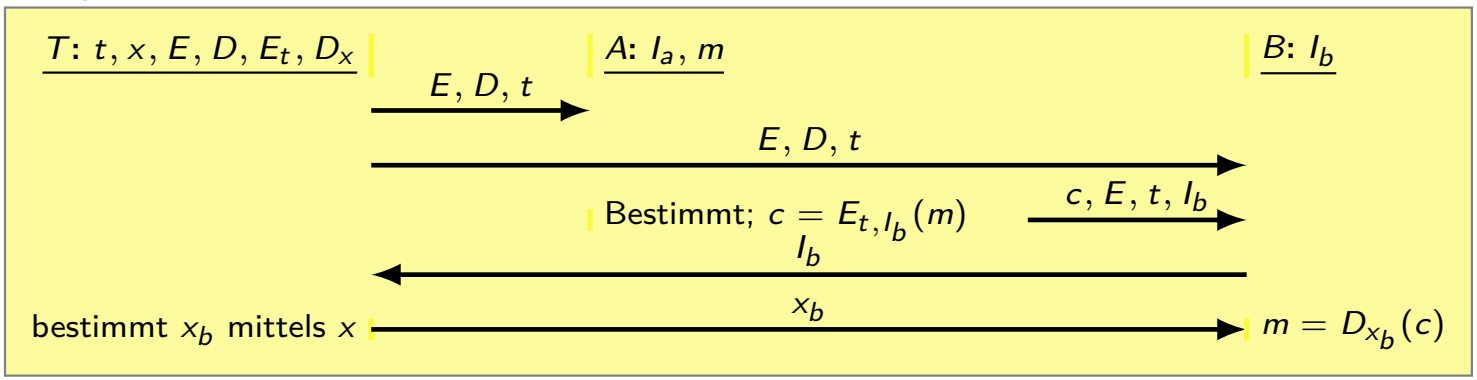
- Zusammenfassung:
  - $|Hdr| + |PK| = \Theta(\sqrt{n})$.
  - So sicher wie n-BDHE.
- Verallgemeinerungen und offene Fragen:
  - Es gibt Variante des Protokolls, die sicher bzgl. Chosen-Ciphertext-Angriffen ist.
  - Es gibt Variante des Protokolls mit Traitor Tracing: Wird ein illegaler Dekoder gefunden, kann rekonstruiert werden, wer kooperiert hat, um ihn zu bauen.
  - Gibt es ein Verfahren mit kurzen privaten Schlüsseln, bei dem die Größe von Header und Public Key kleiner als $\Omega(\sqrt{n})$ ist?

Einleitung

- Ziel: die Identität soll der öffentliche Schlüssel sein:
- Vorteile:
  - Verschlüsselung "sofort" möglich.
  - Verschlüsselung bevor der Empfänger einen geheimen Schlüssel hat.
  - Verschlüsselung an einen "Unbekannten" möglich: $c = E_{LeiterZPA}(m)$.
  - Verschüssleung mit zeitlicher Einschränkung: $c = E_{LeiterZPA10.2010}(m)$.
  - Einfache Verwaltung der öffentlichen Schlüssel.
  - Hierarchische Variante möglich.
- Nachteile:
  - Man braucht eine vertrauenswürdige Stelle, die eine Art privaten Schlüssel vorbereitet.
  - D.h. der Leiter des ZPA geht zum Rektor und bittet um: $D_{LeiterZPA}$.
- Sicherer Kanal notwendig.

Geschichte und Idee

- Geschichte:
  - 1984: Problemstellung durch Shamir.
  - 2001: Erste Lösung durch Boneh Franklin.
  - 2003: Effizient und sicher durch Boneh Franklin.
  - 2004: Verbesserte Sicherheit durch Boneh Boyen.
  - 2005: Effizient und verbesserte Sicherheit durch Waters.
  - 2005: Hierarchische Variante.
- Möglicher Ablauf:



- Finales System ist hierarchisch.
  - Teilnehmer auf Stufe 0 besitzt Master-Key.
  - Identität eines Teilnehmers auf Stufe l ist: $i_1, i_2, \ldots, i_l$ für $1 \leqslant i \leqslant k$.

Bilineare Abbildungen

Bilineare Abbildung

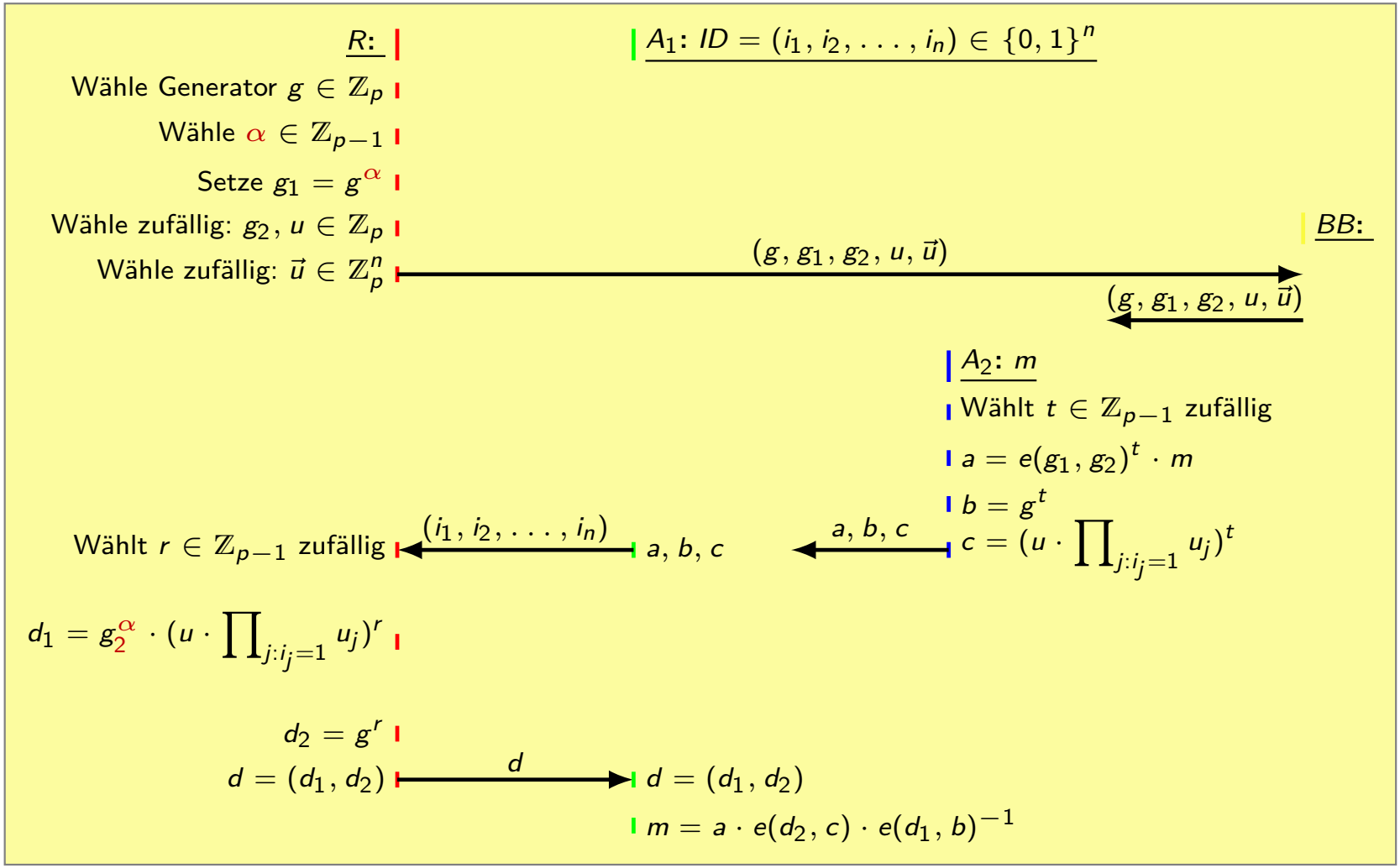
Seien G und G' zyklische Gruppen der Ordnung p für eine Primzahl p. Sei g ein Generator von G. Eine Abbildung $e\colon G \times G \to G'$ nennen wir im Folgenden bilinear, wenn gilt:

1. Bilinear: $\forall u, v \in G, \forall a, b \in \mathbb{Z} : e(u^a, v^b) = e(u, v)^{ab}$.
2. Nicht degeneriert: $e(g, g) \neq 1$.
3. Effizient: e muss effizient berechenbar sein.

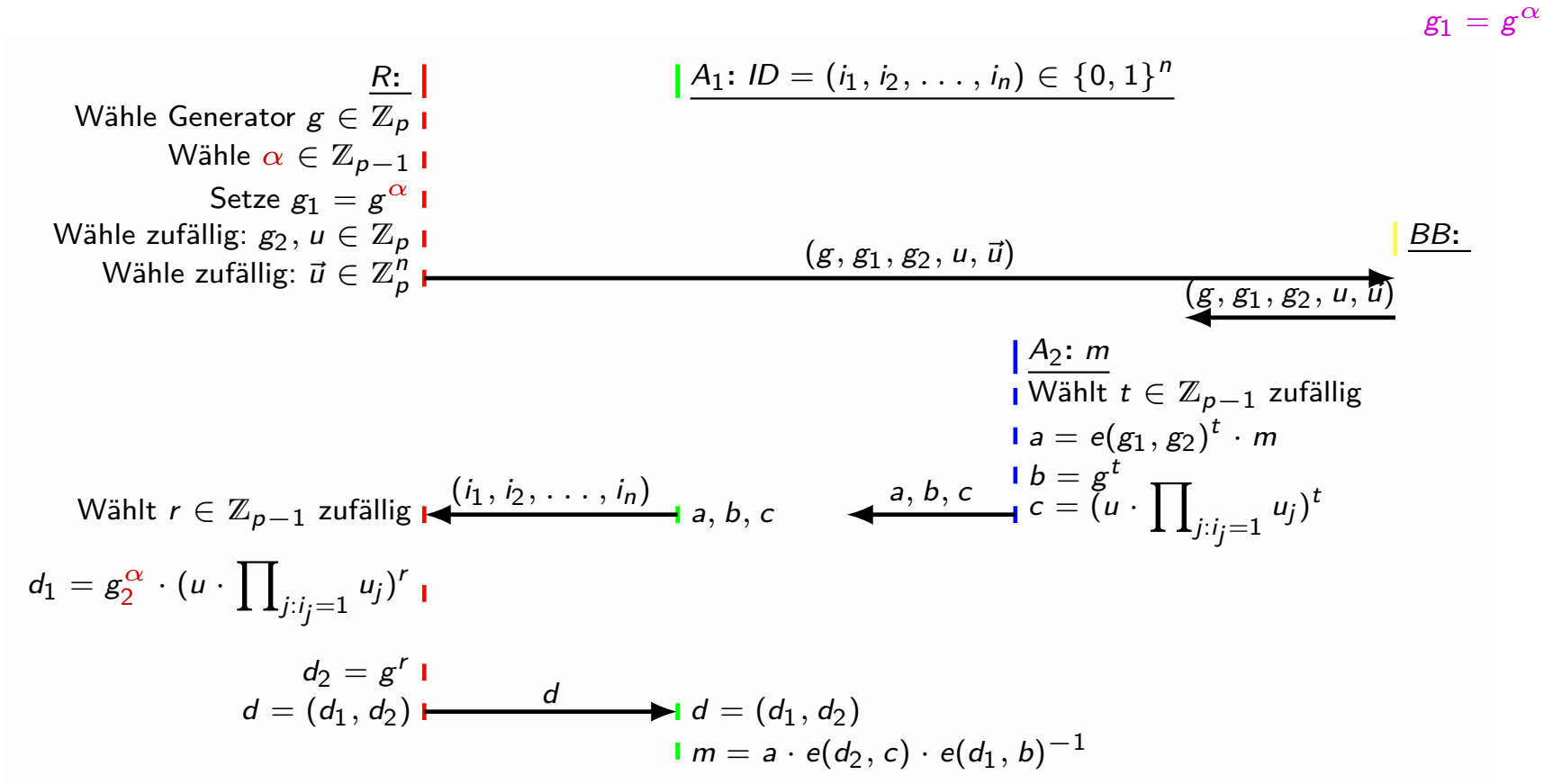
Eigenschaften:

- symmetrisch: $e(u, v) = e(v, u)$.
- distributiv: $e(u, vv') = e(u, v) \cdot e(u, v')$.
- $e(u^a, v) = e(u, v^a)$.

Einfaches System



Korrektheit der Entschlüsselung



- Verschüsselung: $a = e(g_1, g_2)^t \cdot m$, $b = g^t$ und $c = (u \cdot \prod_{j:i_j=1} u_j)^t$.
- Schlüssel: $d_1 = g_2^\alpha \cdot (u \cdot \prod_{j:i_j=1} u_j)^r$ und $d_2 = g^r$.
- Entschüsselung: $m' = a \cdot e(d_2, c) \cdot e(d_1, b)^{-1}$

Korrektheit der Entschlüsselung

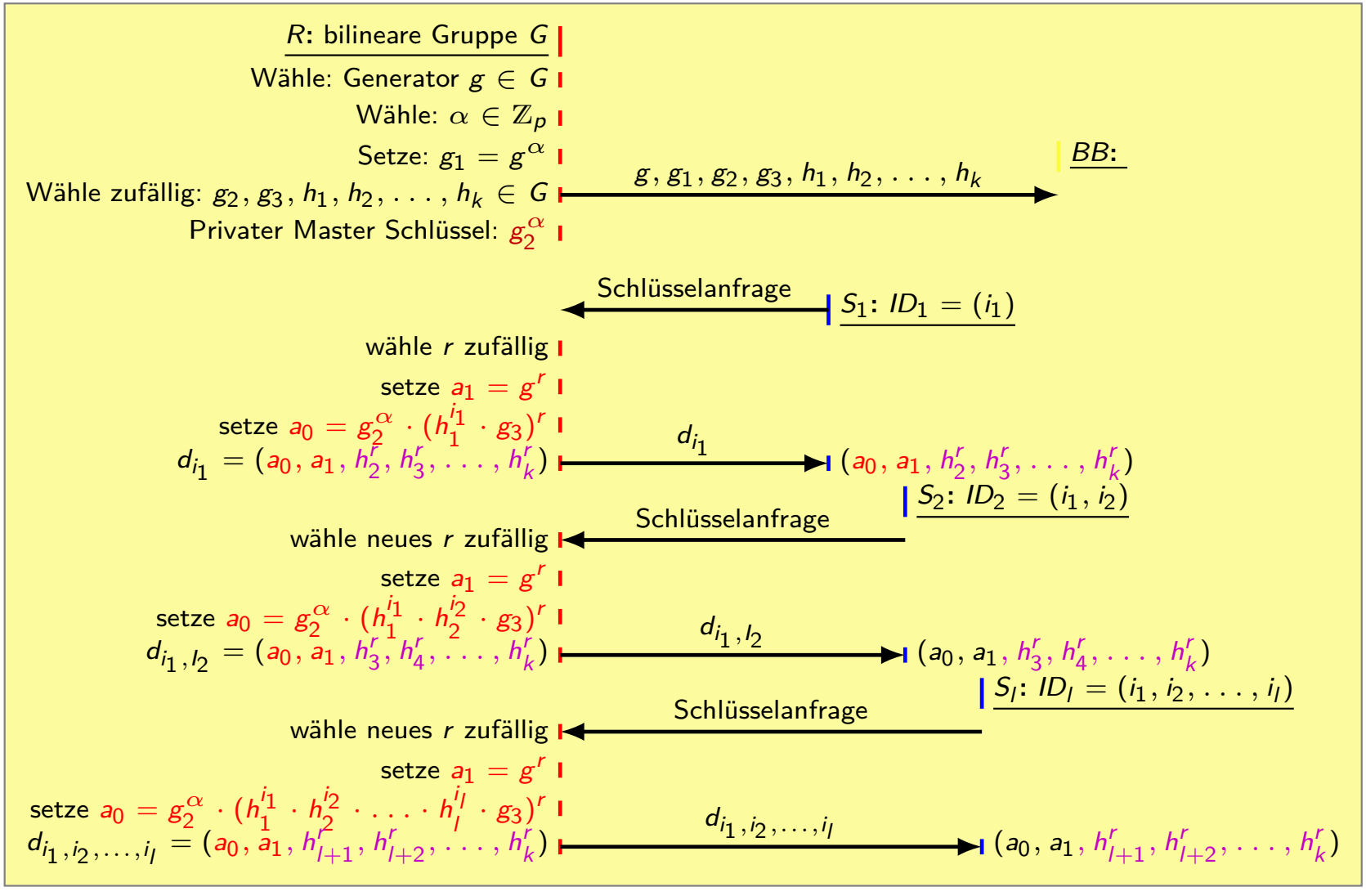
$g_1 = g^\alpha$

- Verschüsselung: $a = e(g_1, g_2)^t \cdot m$, $b = g^t$ und $c = (u \cdot \prod_{j:i_j=1} u_j)^t$.
- Schlüssel: $d_1 = g_2^\alpha \cdot (u \cdot \prod_{j:i_j=1} u_j)^r$ und $d_2 = g^r$.
- Entschüsselung: $m' = a \cdot e(d_2, c) \cdot e(d_1, b)^{-1}$

$$
\begin{aligned}
m' &= a \cdot e(d_2, c) \cdot e(d_1, b)^{-1} \\
&= e(g_1, g_2)^t \cdot m \cdot \frac{e(g^r, (u \cdot \prod_{j:i_j=1} u_j)^t)}{e(g_2^\alpha \cdot (u \cdot \prod_{j:i_j=1} u_j)^r, g^t)} \\
&= e(g_1, g_2)^t \cdot m \cdot \frac{e(g, (u \cdot \prod_{j:i_j=1} u_j)^{rt})}{e(g_2^\alpha, g^t) \cdot e((u \cdot \prod_{j:i_j=1} u_j)^r, g^t)} \\
&= \frac{e(g_1, g_2)^t}{e(g_2^\alpha, g^t)} \cdot m \cdot \frac{e(g, (u \cdot \prod_{j:i_j=1} u_j)^{rt})}{e((u \cdot \prod_{j:i_j=1} u_j)^{rt}, g)} \\
&= \frac{e(g_1, g_2)^t}{e(g_2^\alpha, g)^t} \cdot m \\
&= \frac{e(g_1, g_2)^t}{e(g_2, g_1)^t} \cdot m \\
&= m
\end{aligned}
$$

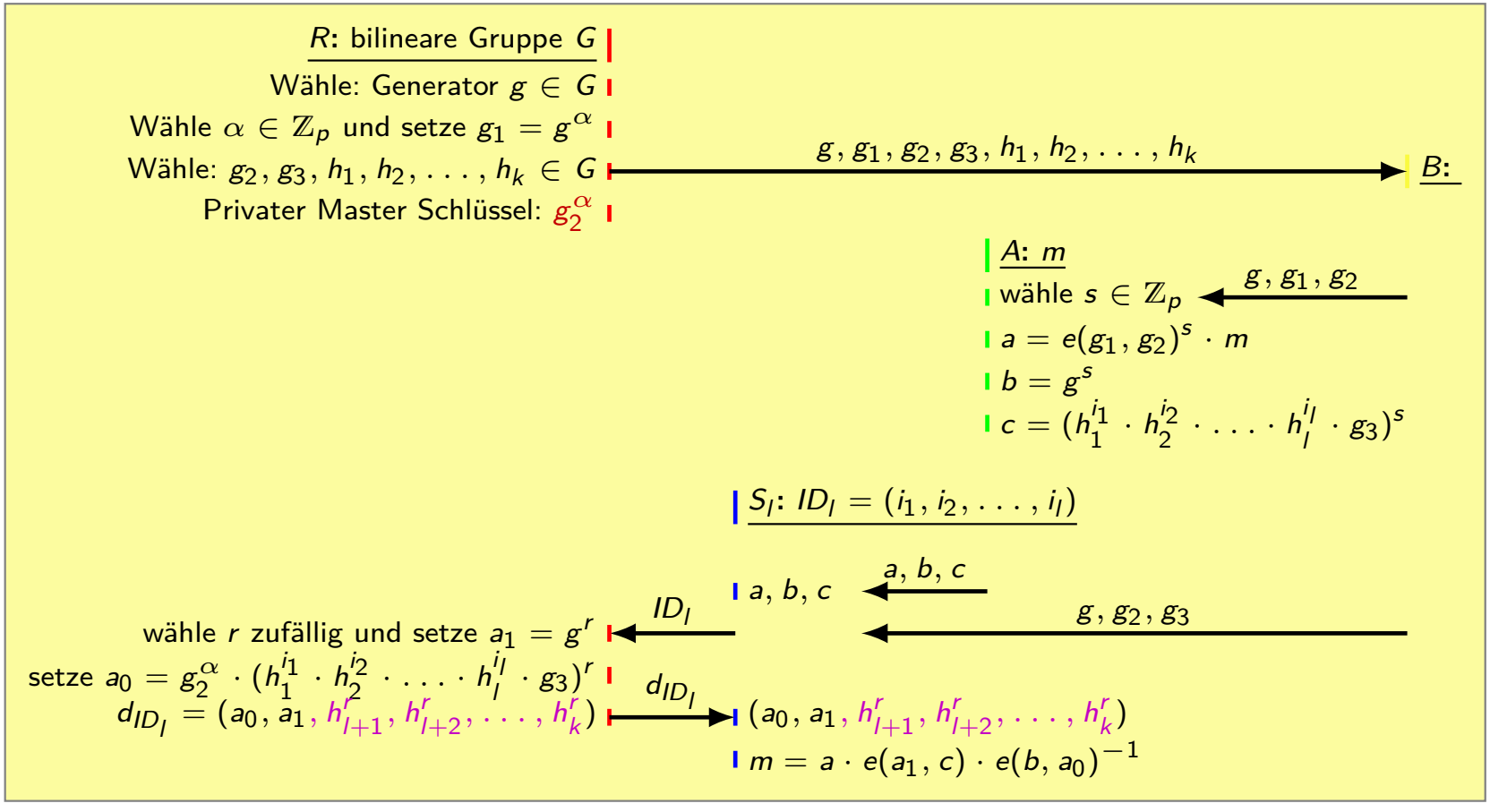
Der Systemaufbau

$g_1 = g^\alpha$



Das System

$g_1 = g^\alpha$



Korrektheit der Entschlüsselung

$g_1 = g^\alpha$

S_I: $ID_I = (i_1, i_2, \ldots, i_l), g, g_2, g_3, a_0, a_1$

mit: $a_0 = g_2^\alpha \cdot (h_1^{i_1} \cdot h_2^{i_2} \cdot \ldots \cdot h_l^{i_l} \cdot g_3)^r$ und: $a_1 = g^r$

$m = a \cdot e(a_1, c) \cdot e(b, a_0)^{-1}$

a, b, c

A: m, g, g_1, g_2

wähle $s \in \mathbb{Z}_p$

$a = e(g_1, g_2)^s \cdot m$

$b = g^s$

$c = (h_1^{i_1} \cdot h_2^{i_2} \cdot \ldots \cdot h_l^{i_l} \cdot g_3)^s$

$$
\begin{aligned}
a \cdot \frac{e(a_1,c)}{e(b,a_0)} &= e(g_1,g_2)^s \cdot m \cdot \frac{e(g^r,(h_1^{i_1}\cdot h_2^{i_2}\cdot\ldots\cdot h_l^{i_l}\cdot g_3)^s)}{e(g^s,g_2^\alpha\cdot(h_1^{i_1}\cdot h_2^{i_2}\cdot\ldots\cdot h_l^{i_l}\cdot g_3)^r)} \\
&= e(g_1,g_2)^s \cdot m \cdot \frac{e(g^r,(h_1^{i_1}\cdot h_2^{i_2}\cdot\ldots\cdot h_l^{i_l}\cdot g_3)^s)}{e(g^s,g_2^\alpha)\cdot e(g^s,(h_1^{i_1}\cdot h_2^{i_2}\cdot\ldots\cdot h_l^{i_l}\cdot g_3)^r)} \\
&= \frac{e(g_1,g_2)^s}{e(g^s,g_2^\alpha)} \cdot m \cdot \frac{e(g^r,(h_1^{i_1}\cdot h_2^{i_2}\cdot\ldots\cdot h_l^{i_l}\cdot g_3)^s)}{e(g^s,(h_1^{i_1}\cdot h_2^{i_2}\cdot\ldots\cdot h_l^{i_l}\cdot g_3)^r)} \\
&= \frac{e(g_1,g_2)^s}{e(g^s,g_2^\alpha)} \cdot m \cdot \frac{e(g,(h_1^{i_1}\cdot h_2^{i_2}\cdot\ldots\cdot h_l^{i_l}\cdot g_3))^{rs}}{e(g,(h_1^{i_1}\cdot h_2^{i_2}\cdot\ldots\cdot h_l^{i_l}\cdot g_3))^{rs}} \\
&= \frac{e(g_1,g_2)^s}{e(g^\alpha,g_2)^s} \cdot m \\
&= \frac{e(g_1,g_2)^s}{e(g_1,g_2)^s} \cdot m \\
&= m
\end{aligned}
$$

Hierarchisches System

- Privater Schlüssel auf Tiefe l:

$$d_{ID_l} = (g_2^{\alpha} \cdot (h_1^{i_1} \cdot h_2^{i_2} \cdot \ldots \cdot h_l^{i_l} \cdot g_3)^r, g^r, h_{l+1}^r, \ldots, h_k^r) = (a_0, a_1, b_{l+1}, \ldots, b_k)$$

- Nun erzeugen wir damit einen privaten Schlüssel der Tiefe $l+1$.
- Wähle s zufällig und setze:

$$\begin{aligned} d_{ID_{l+1}} &= (a_0 \cdot (b_{l+1}^{i_{l+1}} \cdot h_1^{i_1} \cdot \ldots \cdot h_l^{i_l} \cdot g_3)^s, a_1 \cdot g^s, b_{l+2} \cdot h_{l+2}^s, \ldots, b_k \cdot h_k^s) \\ &= (g_2^{\alpha} \cdot (h_1^{i_1} \cdot \ldots \cdot h_l^{i_l} \cdot h_{l+1}^{i_{l+1}} \cdot g_3)^{r+s}, g^{r+s}, h_{l+2}^{r+s}, \ldots, h_k^{r+s}) \end{aligned}$$

- Damit kann ein Teilnehmer auf Tiefe l einen Nachfolger auf Tiefe $l+j$ seinen privaten Schlüssel erzeugen.
- Anwendungen: Verschlüsselung in die Zukunft.
- Sicherheitsbeweis analog wie beim Broadcast möglich.

Literatur

- Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys
  - Dan Boneh, Craig Gentry, and Brent Waters
  - In Proc. of the 25th Annual International Cryptology Conference (CRYPTO), (Santa Barbara, California, USA), August 14-18, 2005, Pages 258–275.
  - Online verfügbar unter: `http://eprint.iacr.org/2005/018.pdf`

Fragen

- Welche Fallen gibt es bei Quittungen?
- Wie arbeitet das Geburtstagsprotokoll?
- Wie sind die Ideen zum Schlüsselaustausch?
- Wie geht der Schlüsselaustausch nach ElGamal?
- Wie ist die Idee zur Zufallszahlengenerierung?
- Wie ist die Idee zur Kartenverteilung?
- Welche Arten der vergesslichen Übertragung gibt es?
- Welche Protokolle zur vergesslichen Übertragung gibt es?
- Was ist die Idee für das Broadcast Protokoll?
- Was ist die Idee von IBE?

Legende

n	Nicht relevant
g	Grundlagen, die implizit genutzt werden
i	Idee des Beweises oder des Vorgehens
s	Struktur des Beweises oder des Vorgehens
w	Vollständiges Wissen