Seminar – Kryptographie

TU-Berlin WS 05/06

Prof. Heß, Dr. Kühn, Prof. Pohst

Secure Delegation of Elliptic-Curve pairing

Ein Verfahren von

B. Chevallier-Mames, J. Coron,

N. McCullagh, D. Naccache, M. Scott

Vortrag: Bernd Hein

bernd.hein@gmx.de

Überblick

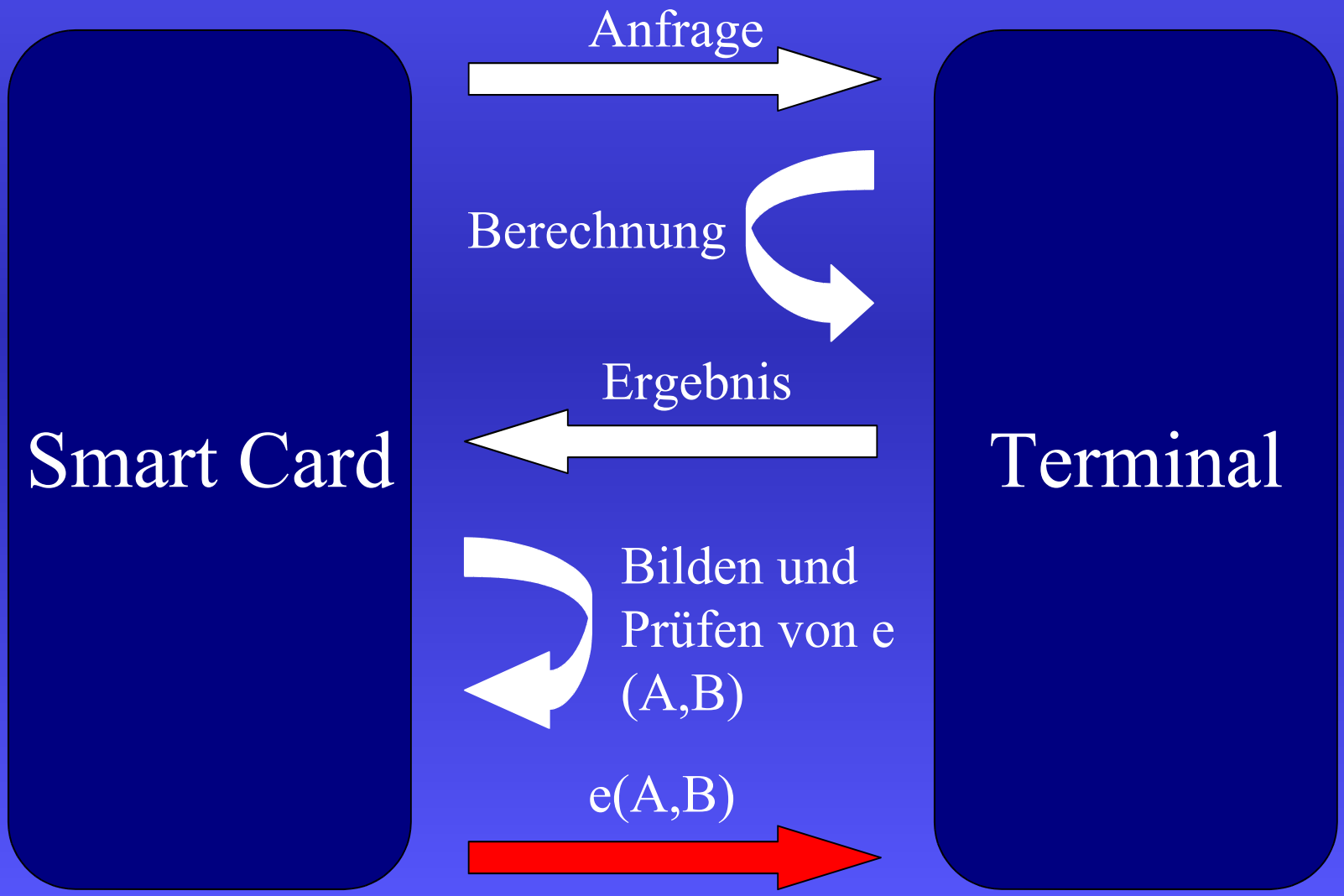
- Einführung
- Grundlagen
- Allgemeines Protokoll
- Effiziente Protokolle

Einführung

- Rechenschwaches Gerät (Smart Card) lässt sich von rechenstärkerem Terminal (PC) indirekt eine Paarung e(A,B) berechnen
- Terminal erfährt nichts über A, B
- Terminal erkennt falsches e(A,B)

- Möglicher Einsatz in Kryptografieverfahren die mit Pairing arbeiten

Einführung


Anfrage
Berechnung
Ergebnis
Smart Card
Terminal
Bilden und
Prüfen von e
(A,B)
e(A,B)

Grundlagen

- Bilineare Abbildungen
- Computational Indistinguishability
- Secure Pairing Delegation

Bilineare Abbildungen

- $\mathcal{G}1$, $\mathcal{G}2$ – additive zyklische Gruppen von primer Ordnung p
- G1 ist Erzeuger von $\mathcal{G}1$, G2 ist Erzeuger von $\mathcal{G}2$
- $\mathcal{G}$T- multiplikative zyklische Gruppe von primer Ordnung p
- bilineare Abbildung e : $\mathcal{G}1 \times \mathcal{G}2 \longrightarrow \mathcal{G}$T
- e(a • U,b • V) = e(U,V)^(ab),
  $\forall$ U$\in \mathcal{G}1$, V $\in \mathcal{G}2$ und a,b $\in \mathbb{Z}$

Computational Indistinguishability

- Unter computational indistinguishability versteht man, daß kein Algorithmus A existiert der in polynomial Zeit entscheiden kann, welche von zwei Ansichten (Wertepaare) welche ist
- Genauer, es nur mit beliebig kleiner Wahrscheinlichkeit kann

Secure Pairing Delegation

Ein Protokoll ist sicher wenn folgende Bedingungen gelten:

- Completeness
- Secrecy
- Correctness

Annahmen:

- Terminal hat kein Geheimnis
- Nur das Terminal kann korrupt sein

Secure Pairing Delegation

Ein Protokoll ist sicher wenn folgende Bedingungen gelten:

- Vollständigkeit (Completeness)
  - Smart Card erhält stets ein korrektes e(A,B), wenn das Terminal nicht korrupt ist
- Geheimhaltung
- Korrektheit

Secure Pairing Delegation

Ein Protokoll ist sicher wenn folgende Bedingungen gelten:

- Vollständigkeit
- Geheimhaltung (Secrecy)
  - Terminal erfährt nichts über A,B, auch wenn es korrupt ist
  - S ≡ ViewT(A,B), die Sicht vom Terminal auf A, B ist nicht von dem eines Simulators mit rein zufälligen Werten unterscheidbar
- Korrektheit

Secure Pairing Delegation

Ein Protokoll ist sicher wenn folgende Bedingungen gelten:

- Vollständigkeit
- Geheimhaltung
- Korrektheit (Correctness)
  - Die Smart Card erkennt ein (gewollt) falsches e (A,B), außer mit vernachlässigbarer Wahrscheinlichkeit, auch wenn das Terminal korrupt ist

Protocol for Secure Delegation of EC pairing

- Einfaches Beispiel mit Fehlannahme
- Verfahrensprotokoll
- Aufwand
- Beweis

Einfaches Beispiel

- Smart Card generiert $x,y \in \mathbb{R}$ zufällig
- Terminal soll $\alpha = e(x \cdot A, y \cdot B)$ berechnen
- Smart Card errechnet

  $eAB = \alpha^{(1/(x \cdot y))}$,

  da $\alpha^{(1/(x \cdot y))} = e(x \cdot A, y \cdot B)^{(1/(x \cdot y))}$
  $= e(A,B)^{((x \cdot y)/(x \cdot y))} = e(A,B)$

Fehlannahme

- Terminal erfährt nichts über A oder B
- Aber wenn Terminal statt e(A,B) nun e(A,B)^r zurückliefert, kann die Smart Card dies nicht entdecken

- Somit ist zwar die Completeness und Secrecy Bedingung erfüllt, nicht aber die Correctness Bedingung für ein Secure Pairing Delegation Protokoll

Verfahrensprotokoll

- Vorgaben
- Anfrage erzeugen
- Anfrage behandeln
- Anfrage überprüfen

Vorgaben

- Smart Card und Terminal kennen beide:
  - $\mathcal{G}1$, $\mathcal{G}2$ – additive zyklische Gruppen der Ordnung p
  - G1, G2 – Erzeuger der Gruppen
  - $\mathcal{G}T$ – multiplikative zyklische Gruppe der Ordnung p
  - bilineare Abbildung e : $\mathcal{G}1 \times \mathcal{G}2 \longrightarrow \mathcal{G}T$
  - e(G1,G2) als Konstante

Anfrage erzeugen

- 0. Smart Card erhält die Punkte A,B
- 1. Smart Card erzeugt zufällig ein
  g1 und g2 $\in \mathbb{Z}p / \mathbf{p}\mathbb{Z}$
- 2. Stellt dann Anfrage nach
  $\alpha 1 = e(A + g1 \bullet G1, G2)$,
  $\alpha 2 = e(G1, B + g2 \bullet G2)$,
  $\alpha 3 = e(A + g1 \bullet G1, B + g2 \bullet G2)$,
  indem A+g1•G1 und B+g2•G2 auf der Smart Card berechnet werden und an das Terminal gesandt werden

Anfrage behandeln

- 3. Terminal berechnet $\alpha 1$, $\alpha 2$, $\alpha 3$ anhand der übermittelten Parameter gemäß der bilinearen Abbildung e
- 4. Terminal sendet $\alpha 1$, $\alpha 2$, $\alpha 3$ an die Smart Card

Anfrage überprüfen

- 5. Smart Card prüft ob $\alpha 1, \alpha 2, \alpha 3 \in \mathcal{G}$ T, indem $\alpha_i \hat{} p = 1$ für i = 1,2,3 gelten muß
- 6. Smart Card berechnet
  $eAB = \alpha 1\hat{}\text{-}g2 \bullet \alpha 2\hat{}\text{-}g1 \bullet \alpha 3 \bullet e(G1,G2)\hat{}(g1 \bullet g2)$
- 7. Smart Card generiert zufällig $a1,r1,a2,r2 \in \mathbb{Z}p$ und fordert vom Terminal die Paarung
  $\alpha 4 = e(a1 \bullet A + r1 \bullet G1, a2 \bullet B + r2 \bullet G2)$
- 8. Smart Card berechnet selbst $\alpha 4'$ mit

  $\alpha 4' = eAB\hat{}\ (a1 \bullet a2) \bullet \alpha 1\hat{}(a1r2) \bullet \alpha 2\hat{}(a2r1)$

  $\bullet\ e(G1,G2)\hat{}(r1r2\text{-}a1g1r2\text{-}a2g2r1)$
- 9. Wenn $\alpha 4' = \alpha 4 \Longrightarrow$ eAB offenbar korrekt und wird ausgeben, sonst HALT

Aufwand

- One-round protocol
  - Für die Berechnung von eAB reicht eine Anfrage mit allen Parametern
- Benötigte Operationen
  - 4 Skalarmultiplikationen in $\mathcal{G}1$ und $\mathcal{G}2$
  - 10 Exponentiationen in $\mathcal{G}T$

Beweis

- Für den Beweis, daß das Protokoll ein Secure Pairing Delegatio Protokoll ist, müssen einzeln die Bedingungen für solche Protkolle geprüft werden
  - Vollständigkeit
  - Geheimhaltung
  - Korrektheit

Beweis – Vollständigkeit

- Für die Vollständigkeit wird einmal gezeigt das eAB = e(A,B) wirklich gilt
- eAB = $\alpha 1^{-g2} \cdot \alpha 2^{-g1} \cdot \alpha 3$
  $\cdot e(G1,G2)^{(g1 \cdot g2)}$
- Wird durch Umstellungen und Einsetzung von $\alpha 1, \alpha 2, \alpha 3$ erreicht

Beweis – Vollständigkeit

- Weiterhin wird gezeigt das $\alpha 4 = \alpha 4'$
- Die Formel $\alpha 4$ mit der das Terminal arbeitet kann in $\alpha 4'$ überführt werden
- So daß Smard Card und Terminal, wenn beide korrekt arbeiten den gleichen Wert berechnen

Beweis – Geheimhaltung

- Für die Geheimhaltung muß gezeigt werden, daß $S \equiv ViewT(A,B)$
- Terminal erhält von Smard Card folgende Tupel: $(\alpha1, \alpha2, \alpha3, \alpha4) = (A + g1 \bullet G1, B + g2 \bullet G2, a1 \bullet A + r1 \bullet G1, a2 \bullet B + r2 \bullet G2)$
- In allen ist eine zufällige Komponente enthalten, wodurch in der zyklischen Gruppe kein Rückschluß auf A oder B möglich ist
- Die Geheimhaltungseigenschaft ist erfüllt, da das Terminal nur zufällige, unabhängig verteilte Punkte innerhalb der Gruppe erhält

Beweis – Korrektheit

- Für die Korrektheit muß gezeigt werden, daß die Wahrscheinlichkeit, daß die Smard Card ein falsches eAB ausgibt, vernachlässigbar gering ist

Beweis – Korrektheit

- 4 Fälle:
  - eAB korrekt, Ausgabe erfolgt
  - eAB korrekt, Ausgabe erfolgt nicht
  - eAB falsch, Ausgabe erfolgt
  - eAB falsch, Ausgabe erfolgt nicht

Beweis – Korrektheit

- 4 Fälle:
  - eAB korrekt, Ausgabe erfolgt
  - eAB korrekt, Ausgabe erfolgt nicht
  - **eAB falsch, Ausgabe erfolgt**
  - eAB falsch, Ausgabe erfolgt nicht
- gesucht P(Ausgabe erfolgt | eAB falsch)

Beweis – Korrektheit

- Bedingung das der Fall eintritt:

  $\alpha 4 = \alpha 4'$,

  obwohl Terminal in $\alpha 1,2,3$ geschummelt hat
- Das Terminal kennt aber nicht die Parameter mit denen die Smard Card $\alpha 4'$ berechnet
- Die Wahrscheinlichkeit richtig zu raten wäre 1 / p, wobei p die Gruppenordnung ist
- ...(Rest siehe Tafel)

Effizientere Protokollvarianten

- Die Verfahren stellen jeweils Bedingungen an die Parameter A, B und arbeiten alle mit dem Boneh und Franklin Verfahren
- Public B
- Public A und B
- Constant Point
  - Constant A and public A, B
  - Constant A and public B

Boneh und Franklin‘s IBE

- Pairing, bilineare Abbildung
- Arbeitet mit Punkten auf Elliptischer Kurve
- Sender generiert Session key zum Verschlüsseln g =e(Q, xP)^r
  - Q – ist Public Key
  - P, xP – Systemparameter, r – zufällig gewählt
- rP wird dem Empfänger geschickt
- Empfänger beantragt vom Private Key Generator seinen Private Key xQ
- e(xQ, rP) = e(Q, xP)^r = g

Public B

- Beim Entschlüsseln mit dem Boneh and Franklin IBE Verfahren, ist der Punkt A der private Schlüssel des Benutzers und B ein Teil des verschlüsselten Textes
- Deshalb muss B nicht geheim sein

Public B

- Selbes Protokoll wie im allgemeinen Fall, nur da B nicht geheim bleiben muss, kann g2 = 0 gesetzt werden

  $\alpha 1 = e(A + g1 \bullet G1, G2)$,

  $\alpha 2 = e(G1, B)$,

  $\alpha 3 = e(A + g1 \bullet G1, B)$

  $\alpha 4 = e(a1 \bullet A + r1 \bullet G1, a2 \bullet B + r2 \bullet G2)$

Public B

- Terminal berechnet eAB = α2^-g1 • α3
- α4 = α4´ $\Longrightarrow$ eAB korrekt

  α4´= eAB^ (a1•a2) • α1^(a1r2) • α2^(a2r1)

  • e(G1,G2)^(r1r2-a1g1r2)
- Benötigt nur 3 Skalarmultiplikationen in $\mathcal{G}$ 1 und $\mathcal{G}$ 2, sowie 8 Exponentiationen in $\mathcal{G}$ T
- Beweis analog zum allgemeinen Fall

Public A und B

- Beim Verschlüsseln mit dem Boneh and Franklin IBE Verfahren, ist der Punkt A der öffentliche identitätsbasierte Schlüssel des Empfängers und B der öffentliche Schlüssel der Trusted-Party
- Da deshalb A und B schon öffentlich bekannt sind, brauchen sie im Verfahren nicht geheim gehalten werden. Dadurch wird die „Secrecy“ Eigenschaft nicht mehr benötigt

Public A und B

- Protokoll ähnlich zu Public A, nur daß noch g1=0 gesetzt werden kann

  $\alpha 1 = e(A, G2)$,

  $\alpha 2 = e(G1, B)$,

  $\alpha 3 = e(A, B)$

  $\alpha 4 = e(a1 \bullet A + r1 \bullet G1, a2 \bullet B + r2 \bullet G2)$

Public A und B

- Terminal setzt eAB = $\alpha 3$
- $\alpha 4 = \alpha 4' \Longrightarrow$ eAB korrekt

  $\alpha 4' =$ eAB^ (a1•a2) • α1^(a1r2) • α2^(a2r1)

  • e(G1,G2)^(r1r2)
- Benötigt nur 2 Skalarmultiplikationen in $\mathcal{G}$ 1 und $\mathcal{G}$ 2, sowie 7 Exponentiationen in $\mathcal{G}$ T
- Beweis analog zum allgemeinen Fall

Constant Point

- Constant, public A
  und public B
- Constant A
  und public B

Constant, public A und public B

- Beim Verschlüsseln mit dem Boneh and Franklin IBE Verfahren, ist der Punkt A der öffentliche Schlüssel der Trusted-Party und B der öffentliche identitätsbasierte Schlüssel des Empfängers
- A ist konstant
- A und B sind nicht geheim

Constant, public A und public B

- Vorgaben wie im allgemeinen Fall, zusätzlich enthält Smart Card

  ein $Q \in \mathcal{G}2$ und $e(A,Q)$

- Anfrage an Terminal

  $\alpha 1 = e(A, B)$,

  $\alpha 2 = e(A, r \bullet B + Q)$ mit zufälligem $r \in \mathbb{Z}p$

Constant, public A und public B

- Wenn $\alpha 1 \hat{} p = 1$ und $(\alpha 1 \hat{} r) \bullet e(A,Q) = \alpha 2$, dann ist $eAB = \alpha 1 = e(A,B)$
- Benötigt nur 1 Skalarmultiplikation und 2 Exponentiationen in $\mathcal{G}$ T

Constant A und public B

- Beim Entschlüsseln mit dem Boneh and Franklin IBE Verfahren, ist der Punkt A der private Schlüssel des Benutzers und B Teil des verschlüsselten Textes
- A ist konstant
- B ist nicht geheim

Constant A und public B

- Vorgaben wie im allgemeinen Fall, zusätzlich enthält Smart Card

  ein $Q \in \mathcal{G}2$, $e(A,Q)$ und A

- Anfrage an Terminal

  $\alpha 1 = e(x \bullet A, B)$,

  $\alpha 2 = e(y \bullet A, z \bullet (B+Q))$

Constant A und public B

- Smart Card berechnet

  $eAB = \alpha 1^{(1/x)}$

  $\alpha 3 = \alpha 2^{(1/yz)}$
- Wenn $eAB^p = 1$ und

  $eAB \bullet e(A,Q) = \alpha 3 \Longrightarrow$ eAB korrekt

- Benötigt nur 3 Skalarmultiplikationen, und 3 Exponentiationen in $\mathcal{G}$ T

Fazit und Ausblick

- Protokoll erfüllt bedingungslose Sicherheit
- Verzicht auf bedingungslose Sicherheit und Akzeptanz von berechenbarer Sicherheit würde eine Optimierung hinsichtlich der benötigten Rechenoperationen bringen