Authentifizierung

Proseminar/Seminar

Kryptographie und Datensicherheit

SoSe 2009 – Universität Potsdam

Jan Jantzen

Gliederung

- Authentifizierung (Einleitung)
- Challenge-Response-Authentifizierung
  - Secret Key
    - Gegenseitige Authentifizierung
  - Public key
    - Zertifikate
- Schnorr Identification Scheme

Authentifizierung

Authentifizierung ist der Vorgang der Überprüfung einer behaupteten Identität, beispielsweise einer Person oder eines Objekts, wie beispielsweise eines Computersystems.

Dies kann auf drei Wegen (und Kombinationen derer) erreicht werden:

- **Körperliches Merkmal / Biometrie**
  z.B. Aussehen, Fingerabdrücke...
- **Besitz**
  z.B. Ausweis, Reisepass...
- **Wissen**
  z.B. Passwörter, PIN...

Authentifizierung

Typische Szenarien:

- Kreditkarteneinkäufe
- Geld abheben am Bankautomaten
- Remote login

- In der Praxis meist Unsicher Implementiert
- Wenn ein Zuhörer die gesendeten Passwörter abfängt, kann dieser sich mit Hilfe dieser Ausweisen.

Authentifizierung

Das Ziel von Identifikationsprotokollen ist also, zu verhindern, dass ein Lauscher Eve, der mithört wie Alice sich bei Bob authentisiert sich nicht anschließend selber als Alice Ausweisen kann.

Challenge-Response-Authentifizierung

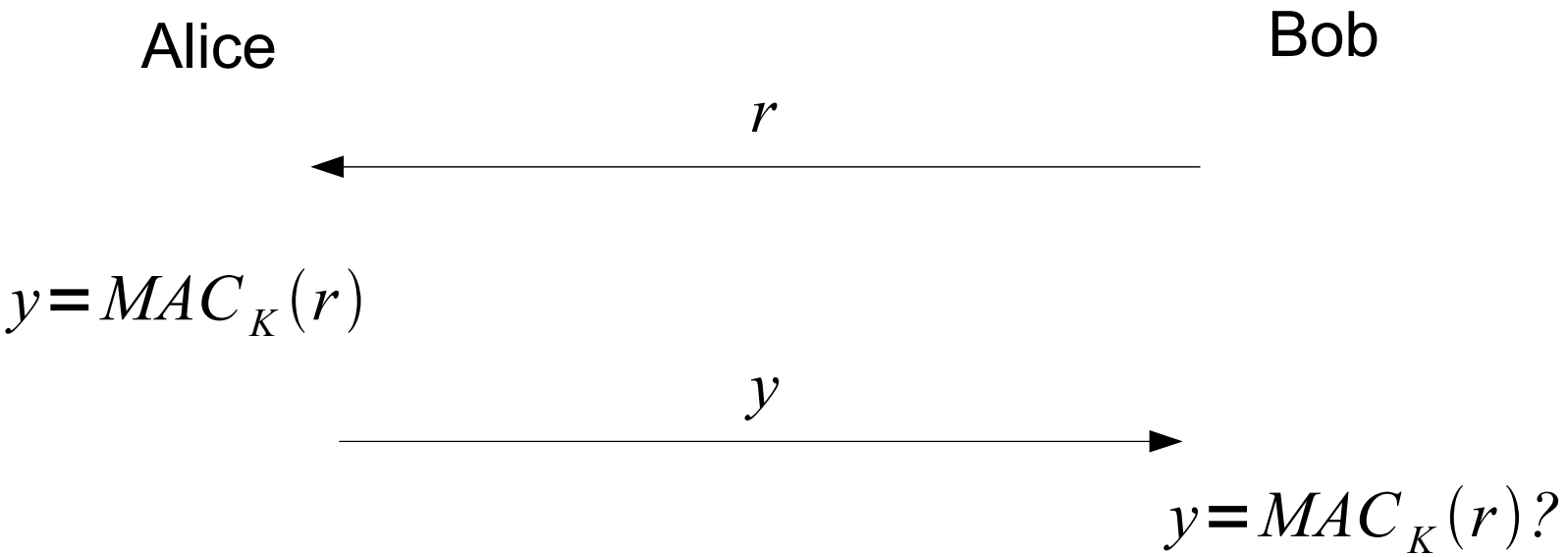
Um dynamische Antworten zu erzeugen werden zufällige Herausforderungen benutzt.

Protokoll 9.1: unsecure Challenge-And-Response

1. Bob erzeugt zufällige Herausforderung r und sendet diese an Alice.
2. Alice berechnet $y = MAC_K(r)$
und sendet y an Bob.
3. Bob berechnet $y' = MAC_K(r)$
Wenn $y = y'$ Akzeptiert Bob, sonst lehnt er ab.

Challenge-Response-Authentifizierung

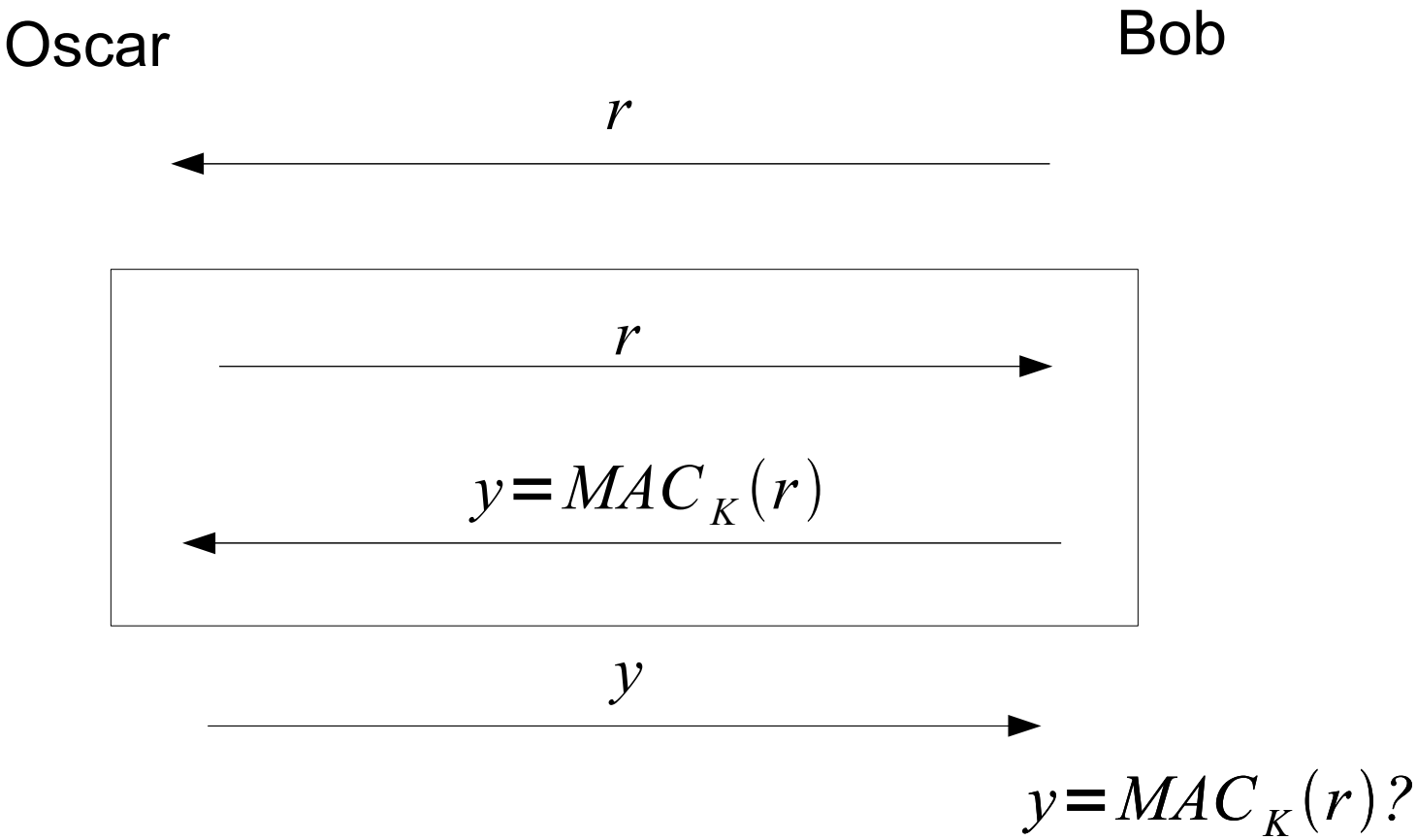
Informationsfluss in Protokoll 9.1



Protokoll 9.1 setzt voraus, dass Alice und Bob einen Schlüssel Ausgetauscht haben und einen Sicheren MAC verwenden.

Challenge-Response-Authentifizierung

Parallel session attack auf Protokoll 9.1



Challenge-Response-Authentifizierung

Protokoll 9.2: (secure) Challenge-And-Response

1. Bob erzeugt zufällige Herausforderung r und sendet diese an Alice.

2. Alice berechnet $y = MAC_K(ID(Alice) \| r)$
und sendet y an Bob.

3. Bob berechnet $y' = MAC_K(ID(Alice) \| r)$
   Wenn $y = y'$ Akzeptiert Bob, sonst lehnt er ab.

Oscar kann die *parallel session attack* nicht erfolgreich auf Protokoll 9.2 anwenden, da er auf die Herausforderung r
$y' = MAC_K(ID(Bob) \| r)$ von Bob als Antwort erhält.

Challenge-Response-Authentifizierung

Zum Beweis der Sicherheit treffen wir folgende Annahmen:

- Der Schlüssel K ist nur Alice und Bob bekannt
- Alice und Bob sind Ehrlich
- Alice und Bob haben perfekte Zufallsgeneratoren zur Erstellung der Herausforderungen
- Es existiert kein (e,Q)-Fälscher für den MAC mit relevantem e oder Q.

Challenge-Response-Authentifizierung

Angriffsmodell:

- Oscar kann mehrere Sitzungen beobachten.
- Oscars Ziel ist es sich bei Alice oder Bob zum Akzeptieren zu bringen, ohne dass deren Gegenüber an der Sitzung teil nimmt.

Challenge-Response-Authentifizierung

- $y = MAC_K(ID(Alice) \| r)$ kann nicht in einer vorherigen Sitzung von Bob erstellt worden sein.
- Alice hat y nur Berechnet, wenn Bob r schon einmal als Herausforderung genutzt hat, die Wahrscheinlichkeit hierfür ist $Q/2^L$ bei Q bisher beobachteten Sitzungen und einer Länge der Herausforderung von L.
- Oscar kann ohne den Schlüssel y nur mit einer Wahrscheinlichkeit kleiner e berechnen.

  Oscar kann sich also nur mit einer Wahrscheinlichkeit kleiner $Q/2^L + e$ als Alice ausgeben.

Gegenseitige Authentifizierung

Protokoll 9.4: (secure) mutual Challenge-And-Response

1. Bob erzeugt zufällige Herausforderung r_1 und sendet diese an Alice.
2. Alice erzeugt zufällige Herausforderung r_2 und berechnet $y_1 = MAC_K(ID(Alice) \| r_1 \| r_2)$ und sendet y_1 und r_2 an Bob.
3. Bob berechnet $y'_1 = MAC_K(ID(Alice) \| r_1 \| r_2)$
   Wenn $y = y'$ Akzeptiert Bob, sonst lehnt er ab.
   Bob berechnet zusätzlich $y_2 = MAC_K(ID(Bob) \| r_2)$ und sendet y_2 an Alice
4. Alice Berechnet $y_2' = MAC_K(ID(Bob) \| r_2)$
   Wenn $y_2 = y'_2$ Akzeptiert Alice, sonst lehnt sie ab.

Public-Key-Authentification

- Notwendig, wenn es keinen vorher vereinbarten Schlüssel gibt.
- Erfordert eine public-key Infrastruktur

  Hier eine Vertrauenswürdige Autorität (VA), die Schlüssel verteilt und signiert.

  Der Schlüssel zur Verifizierung der Signaturen der VA ver_{VA} muss allgemein bekannt sein.

Zertifikate

Zertifikate erlauben eine gegenseitige Verifizierung der Authentizität der öffentlichen Schlüssel.

Bestehen aus:

- Informationen zur Identifizierung
- Öffentlicher Schlüssel
- Signatur der VA

Zeigt nur Authentizität des Schlüssels und beweist nicht die Identität.

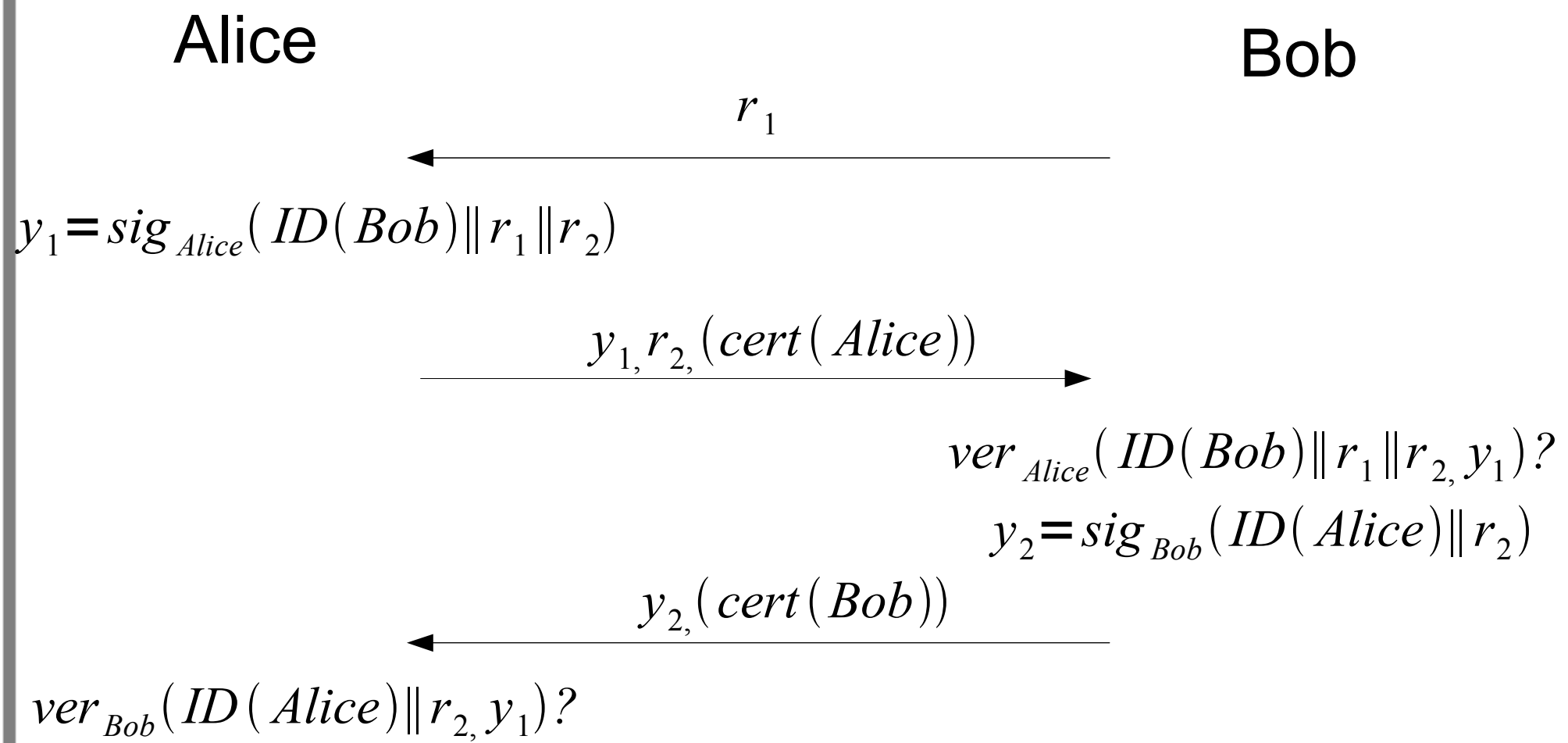
Public-Key-Authentification

Protokoll 9.6: public-key mutual Challenge-And-Response

1. Bob erzeugt zufällige Herausforderung r_1 und sendet diese und $cert(Bob)$ an Alice.
2. Alice erzeugt zufällige Herausforderung r_2 und berechnet $y_1 = sig_{Alice}(ID(Bob) \| r_1 \| r_2)$ und sendet y_1 , r_2 und $cert(Alice)$ an Bob.
3. Bob überprüft $cert(Alice)$ und $ver_{Alice}(ID(Bob) \| r_1 \| r_2, y_1)$ Wenn diese authentisch sind Akzeptiert Bob, sonst lehnt er ab. Bob berechnet zusätzlich $y_2 = sig_{Bob}(ID(Alice) \| r_2)$ und sendet y_2 an Alice
4. Alice überprüft $cert(Bob)$ und $ver_{Bob}(ID(Alice) \| r_2, y_1)$ Wenn diese authentisch sind Akzeptiert Alice, sonst lehnt sie ab.

Public-Key-Authentification

Datenfluss im Protokoll 9.6



Schnorr Identification Scheme

Entwerfen von Identifizierungsverfahren ohne bisherige kryptologische Methoden als Bausteine zu verwenden kann Vorteile in Laufzeit und Menge der zu übertragenen Daten bringen.

Ein solches Verfahren ist das

Schnorr Identification Scheme

Schnorr Identification Scheme

Voraussetzungen:

- $\alpha \, ist \; ein \, Element \; der \, Ordnung \, q \, \text{in} \, der \, Gruppe \, \mathbb{Z}_p^*$
- $p \, ist \;\; prim \, und \;\; p-1 \equiv 0 \, (mod \; q)$
- $\log_\alpha(\beta) \, wird \; definiert \, , \;\; für \;\; jedes \, \beta \in \langle \alpha \rangle$
- $und \; 0 \leq \log_\alpha(\beta) \leq q-1$

VA wählt Systemparameter:

- $p : gro\ss e \; Primzahl \, , \; p \sim 2^{1024}$
- $q : primteiler \; von \; p-1 \quad , q \sim 2^{160}$
- $\alpha \in \mathbb{Z}_p^* \, mit \; Ordnung \; q$

Teilnehmer wählen privaten Schlüssel a und und berechnen öffentl. Schlüssel $v = \alpha^{-a} \; mod \;\; p$

Schnorr Identification Scheme

Protokoll 9.8: Schnorr Identification Scheme

1. Alice erzeugt zufällige Nummer k , $mit\, 0 \le k \le q-1$ und berechnet $\gamma = \alpha^k \ mod\ p.$ Sie sendet Cert(Alice) und γ an Bob.
2. Bob Überprüft v und Cert(Alice). Er erstellt eine zufällige Herausforderung r und sendet diese an Alice.
3. Alice berechnet $y = k + ar \ mod\, q$ und sendet y an Bob
4. Bob überprüft, ob $\gamma \equiv \alpha^y v^r \ (mod\ p)$, wenn ja akzeptiert Bob, sonst lehnt er ab.

Schnorr Identification Scheme

Folgende Kongruenzen zeigen, dass Alice sich gegenüber Bob authentisieren kann.

$$\alpha^{v} v^{r} \equiv \alpha^{\mathrm{k+ar}} v^{r} (mod\ p)$$
$$\equiv \alpha^{\mathrm{k+ar}} \alpha^{-ar} (\mathrm{mod\ p})$$
$$\equiv \alpha^{\mathrm{k}} (mod\ p)$$
$$\equiv \gamma^{\mathrm{k}} (mod\ p)$$

Schnorr Identification Scheme

Beispiel:
Angenommen: $p = 88667, q = 1031$

$\alpha = 70322$ hat Ordnung q in $\mathbb{Z}_p^*$

Alice' geheimer Schlüssel a ist 755

Dann ist $v = \alpha^{-a} \; mod \; p$

$v = 70322^{1031\text{-}755} \; mod \; 88667 = 13136$

Schnorr Identification Scheme

Angenommen Alice
Wählt die zufällige Nummer
K = 543 und berechnet

$$p = 88667, q = 1031$$
$$\alpha = 70322, v = 13136$$

$$\gamma = \alpha^k \; mod \; p$$
$$\gamma = 70322^{543} \; mod \; 88667 = 84109$$

Und sendet γ zu Bob, Bob sendet die Herausforderung r = 1000. Nun berechnet Alice

$$y = k + ar \; mod \; q$$
$$y = 543 + 755 * 1000 \; mod \; 1031 = 851$$

Und sendet y an Bob, welcher Verifiziert, dass

$$84109 \equiv 70322^{851} * 13136^{1000} \; (mod \; 88667)$$

Schnorr Identification Scheme

Sicherheit wird durch funktionale Reduktion auf das Diskrete Logarithmus Problem bewiesen:

Angenommen, es gäbe einen erfolgreichen Betrüger. Dieses kann man nutzen, um aus dem öffentlichen Schlüssel $\gamma = \alpha^k$ den geheimen Schlüssel a zu bestimmen, also den Diskreten Logarithmus a von γ zu berechnen.
Dies steht im Widerspruch zur Annahme, der diskrete Logarithmus sei schwierig.

Schnorr Identification Scheme

Reduktion auf das DLP (Informell):

1. Simuliere den Algorithmus zur Identifikation, speichere den Zustand vor dem Senden der Herausforderung r_1 an den Betrüger.
2. Wiederhole die Simulation an gespeicherten Zustand, wähle ein zufälliges r_2 als Herausforderung

- Seien γ_1 und γ_2 die beiden (verschiedenen) Antworten zum gleichen Zufallswert k.
- Es gilt $\gamma_1 - \gamma_2 = (k + ar_1) - (k + ar_2) = a(r_1 - r_2) \bmod q$, also

$$a = (r_1 - r_2)/(\gamma_1 - \gamma_2) \bmod q$$

Referenzen

- Douglas R. Stinson: Cryptography: Theory and Practice. 3rd Edition, Chapman & Hall/CRC 2006
- Script zur Vorlesung Kryptographie im Wintersemester 2007 an der Technischen Universität Berlin. Dozenten: Prof. Dr. Florian Heß, Dipl.-Math. Osmanbey Uzunkol
  http://www.math.tu-berlin.de/~hess/krypto-ws2007/
- Wiki Projekt Kryptologie
  http://de.wikipedia.org/wiki/Wikipedia:WikiProjekt_Kryptologie

Alle Webadressen stand:05.07.2009