Beweisbar sichere Verschlüsselung

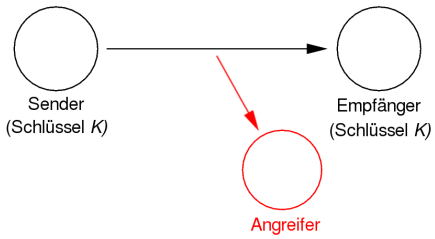
ITS-Wahlpflichtvorlesung

Dr. Bodo Möller
Ruhr-Universität Bochum
Horst-Görtz-Institut für IT-Sicherheit
Lehrstuhl für Kommunikationssicherheit
bmoeller@crypto.rub.de

$Id: bsv.ltx,v 1.62 2007/06/06 15:57:10 bm Exp $

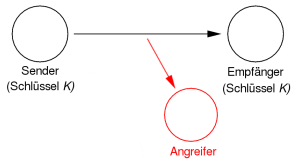
Sicherheit gegen aktive Angreifer

- Wir bleiben bei *symmetrischer Verschlüsselung,* wollen den Angreifern mehr erlauben
- *Bisher* Sicherheitsbegriffe mit *passivem Angreifer:*

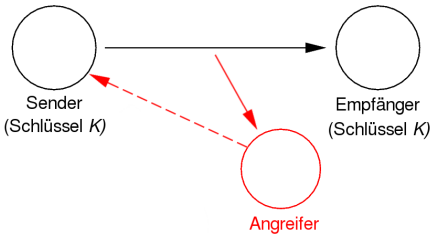


Sicherheit gegen aktive Angreifer (Forts.)

- Passiver Angreifer:



- In den Modellen (RoR-CPA usw.) war der Angreifer kein reiner Beobachter, sondern bekam ein *Verschlüsselungsorakel* verschiedener Art:

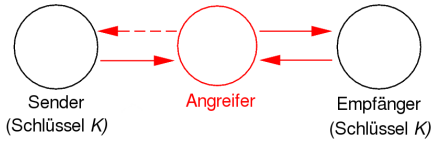


  Sinn dieses Orakels war es, alle denkbaren Gegebenheiten bei der Plaintext-Verteilung abzudecken.

- Deshalb *Chosen Plaintext Attack* (CPA)

Sicherheit gegen aktive Angreifer (Forts.)

- *Jetzt* wollen wir *aktive Angreifer* modellieren, die den Übertragungskanal zwischen Sender und Empfänger kontrollieren:



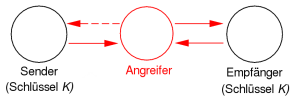
- Der Angreifer hat wie bisher ein *Verschlüsselungsorakel* (er sagt dem Sender, was verschlüsselt werden soll);
  zusätzlich jetzt ein *Entschlüsselungsorakel* (er sieht Entschlüsselungsresultate des Empfängers)

- Deshalb *Chosen Ciphertext Attack* (CCA) – dem Angreifer bleiben dabei alle Möglichkeiten aus dem CPA erhalten!

  (Also eigentlich "chosen plaintext and ciphertext")

Sicherheit gegen aktive Angreifer (Forts.)

- Aktiver Angreifer (CCA):



- Aber wenn der Angreifer ein Entschlüsselungsorakel zur Verfügung hat, was bleibt ihm dann eigentlich noch zu tun ...?
- Für eine sinnvolle Modellierung wird das *Entschlüsselungsorakel eingeschränkt!*
- c ist *als Anfrage ans Entschlüsselungsorakel* nur dann zulässig, wenn c *nicht* eine der *Antworten des Verschlüsselungsorakels* ist von vorher im Angriffsspiel
- So können wir *alle von CPA bekannten Angriffe analog* definieren:

RoR-CCA	LoR-CCA	FtG-CCA	Sem-CCA
RoR-OTCCA	LoR-OTCCA		Sem-OTCCA

- *OTCCA* heißt hier: Einmal*verschlüsselung* mit CCA (Verschlüsselungsorakel einmal, Entschlüsselungsorakel mehrfach zu verwenden!)

Sicherheit gegen aktive Angreifer (Forts.)

- Sem-CCA und Sem-OTCCA *(semantische Sicherheit unter Chosen Ciphertext Attack)* brauchen viele technische Details, werden hier nicht weiter betrachtet!
- Wichtig ist aber: Auch hier wurden Äquivalenzen gezeigt zu den anderen Formalisierungen
- Wir betrachten als detaillierte *Beispiele* RoR-OTCCA und LoR-CCA:
  - ein *Real-or-Random*-Angriffsspiel
  - mit *One-Time*-Verschlüsselungsorakel
  - und mit Entschlüsselungsorakel (für *Chosen Ciphertext Attack*);

  und

  - ein *Left-or-Right*-Angriffsspiel
  - mit uneingeschränktem Verschlüsselungsorakel (d. h. nicht "one time")
  - und mit Entschlüsselungsorakel (*Chosen Ciphertext Attack*)

RoR-OTCCA-Sicherheit

- *Ablauf des RoR-OTCCA-Angriffsspiels*

  mit Fällen $b = 1$ ("real") und $b = 0$ ("random")
  auf ein Verschlüsselungsschema $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ für Angreifer $\boldsymbol{A}$:

  - $K \xleftarrow{\$} \mathcal{K}$
  - $\boldsymbol{A}$ stellt Anfragen c an Orakel $D(\cdot)$, erhält jeweils $\mathcal{D}_K(c)$
  - $\boldsymbol{A}$ stellt *eine* Anfrage m an Orakel $E(\cdot)$, erhält $c_{\text{challenge}} = E_b(m)$,

    wobei $E_1(\cdot)$ "Real"-Verschlüsselungsorakel ist (Antwort $E_K(m)$)
    und $E_0(\cdot)$ "Random"-Verschlüsselungsorakel (Antwort $E_K(m_0)$
    mit $m_0 \in_{\$} \{0,1\}^{|m|}$)
  - $\boldsymbol{A}$ stellt wieder Anfragen c an Orakel $D(\cdot)$;

    Anfrage $c = c_{\text{challenge}}$ ist unzulässig (Antwort $\perp$),
    ansonsten erhält $\boldsymbol{A}$ jeweils $\mathcal{D}_K(c)$
  - $\boldsymbol{A}$ gibt ein Bit $\widetilde{b}$ aus
- Hierbei darf $\boldsymbol{A}$ an $D(\cdot)$ jeweils viele Fragen stellen
  (nacheinander, sie können also von vorherigen Antworten abhängen)

RoR-OTCCA-Sicherheit (Forts.)

- *RoR-OTCCA-Vorteil:*

  Wie gewohnt definiert, nämlich

$$\mathrm{Adv}^{\text{RoR-OTCCA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} = \Pr\left[\boldsymbol{A}^{E_1(\cdot),D(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E_0(\cdot),D(\cdot)} \Rightarrow 1\right]$$

- „$(\mathcal{K}, \mathcal{E}, \mathcal{D})$ ist *RoR-OTCCA-sicher*" heißt (informell):

  Für jeden denkbaren Angreifer $\boldsymbol{A}$
  bleibt der Vorteil $\mathrm{Adv}^{\text{RoR-OTCCA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}}$ verschwindend gering

LoR-CCA-Sicherheit

- *Ablauf des LoR-CCA-Angriffsspiels*
  mit Fällen $b = 1$ ("left") und $b = 0$ ("right")
  auf ein Verschlüsselungsschema $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ für Angreifer $\boldsymbol{A}$:
  - $K \xleftarrow{\$} \mathcal{K}$
  - $\boldsymbol{A}$ stellt in beliebiger Reihenfolge
    - Anfragen (m_1, m_0) (mit $|m_1| = |m_0|$) an Orakel $E(\cdot,\cdot)$,
      erhält jeweils die Antwort $\mathcal{E}_K(m_b)$
    - Anfragen c an Orakel $D(\cdot)$;
      unzulässig, falls c unter den bisherigen Antworten von $E(\cdot,\cdot)$ ist
      (dann Antwort $\perp$),
      sonst erhält $\boldsymbol{A}$ jeweils $\mathcal{D}_K(c)$
  - $\boldsymbol{A}$ gibt ein Bit $\widetilde{b}$ aus
- *LoR-CCA-Vorteil:* Wie gewohnt definiert als

$$\mathrm{Adv}^{\text{LoR-CCA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} \;=\; \Pr\left[\boldsymbol{A}^{E_1(\cdot,\cdot),D(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E_0(\cdot,\cdot),D(\cdot)} \Rightarrow 1\right]$$

LoR-CCA-Sicherheit (Forts.)

- LoR-CCA-Vorteil:
  $\mathrm{Adv}^{\text{LoR-CCA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}} \;=\; \Pr\left[\boldsymbol{A}^{E_1(\cdot,\cdot),D(\cdot)} \Rightarrow 1\right] - \Pr\left[\boldsymbol{A}^{E_0(\cdot,\cdot),D(\cdot)} \Rightarrow 1\right]$
- „$(\mathcal{K}, \mathcal{E}, \mathcal{D})$ ist *LoR-CCA-sicher*“ heißt (informell):
  Für jeden denkbaren Angreifer $\boldsymbol{A}$
  bleibt der Vorteil $\mathrm{Adv}^{\text{LoR-CCA}}_{(\mathcal{K},\mathcal{E},\mathcal{D}),\boldsymbol{A}}$ verschwindend gering

CCA-Sicherheit

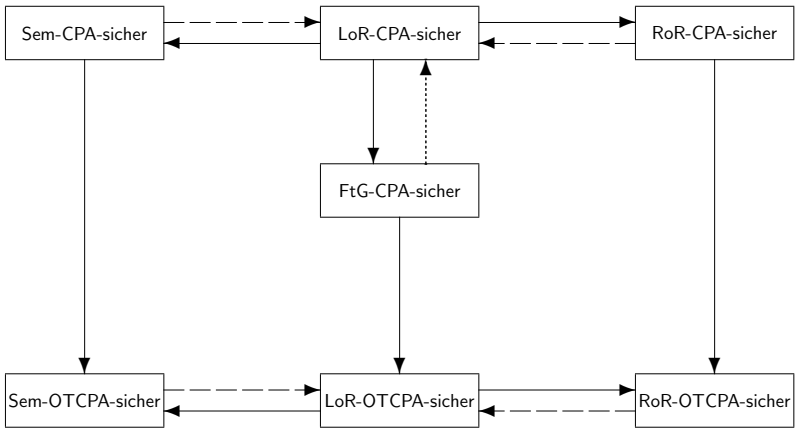
- Diese Angriffsspiele sind, vom Entschlüsselungsorakel $D(\cdot)$ abgesehen, genauso wie die entsprechenden CPA-Angriffsspiele
- ... ein CPA-Angreifer kann also auch als CCA-Angreifer eingesetzt werden (z. B. jeder RoR-OTCPA-Angreifer als RoR-OTCCA-Angreifer)
- Das heißt: *CCA-Sicherheit impliziert CPA-Sicherheit*
  (z. B.: RoR-OTCCA-Sicherheit impliziert RoR-OTCPA-Sicherheit)

CCA-Sicherheit (Forts.)

- Das Entschlüsselungsorakel erlaubt dem Angreifer sehr viel – warum?
  Das Angriffsspiel soll dem Angreifer im Zweifelsfall *mehr* erlauben, als in der Praxis zu befürchten ist:
- CCA *in der Praxis* z. B.:
  Ändere oder ersetze verschlüsselte Daten und beobachte, wie der Empfänger darauf reagiert.
  Das ist kein vollwertiges Entschlüsselungsorakel, liefert aber *partielle Information*
- Wenn sogar für „vollen CCA“ Sicherheit vorliegt, ist man für alle Falle von weiter eingeschränkten Chosen Ciphertext Attacks gut gerüstet!
- Aber Vorsicht: *CCA* (RoR-CCA, LoR-CCA usw.) ist noch *nicht alles,* später kommt noch *Integrity of Ciphertext* hinzu (INT-CTXT)

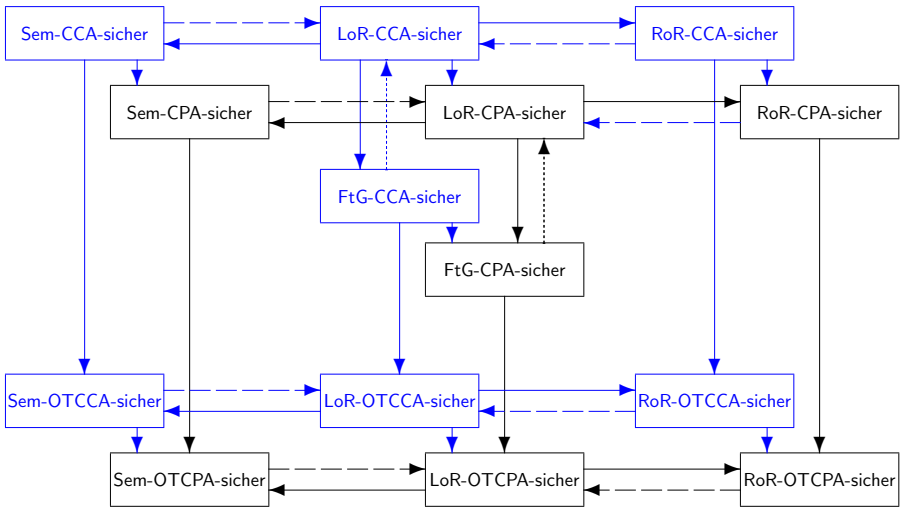
CCA-Sicherheit (Forts.)

- Für CPA-Sicherheitsbegriffe kennen wir folgende Zusammenhänge:



CCA-Sicherheit (Forts.)

- Das wollen wir erweitern mit CCA ...:

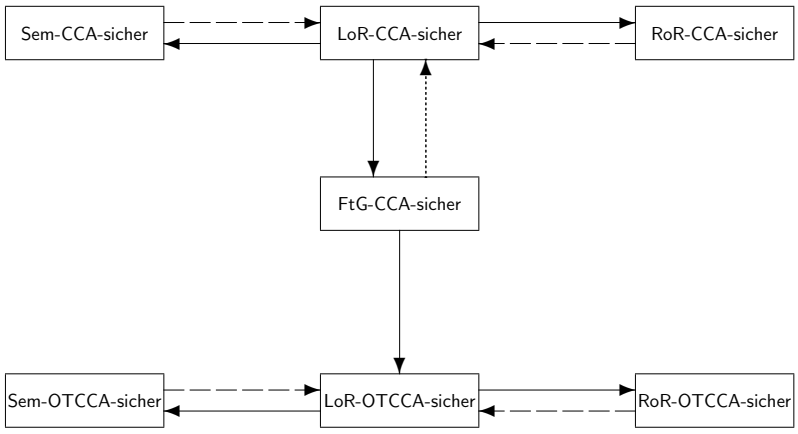


CCA-Sicherheit (Forts.)

- Die Pfeile X-CCA-sicher $\rightarrow$ X-CPA-sicher sind klar
  (X-OTCCA-sicher $\rightarrow$ X-OTCPA-sicher genauso)
- Die Pfeile X-CCA-sicher $\rightarrow$ X-OTCCA sind auch klar
- Diejenigen Pfeile, die nur CPA betreffen, sind von vorher bekannt
- Bleiben anzusehen: ...

CCA-Sicherheit (Forts.)

- Bleiben anzusehen:



- Auf Sem-CCA, Sem-OTCCA werden wir nicht im Detail eingehen!