Unterscheider für DDH durch $\mathcal{A}$

Algorithmus DDH-Unterscheider D

EINGABE: (G, q, g, g^x, g^y, g')

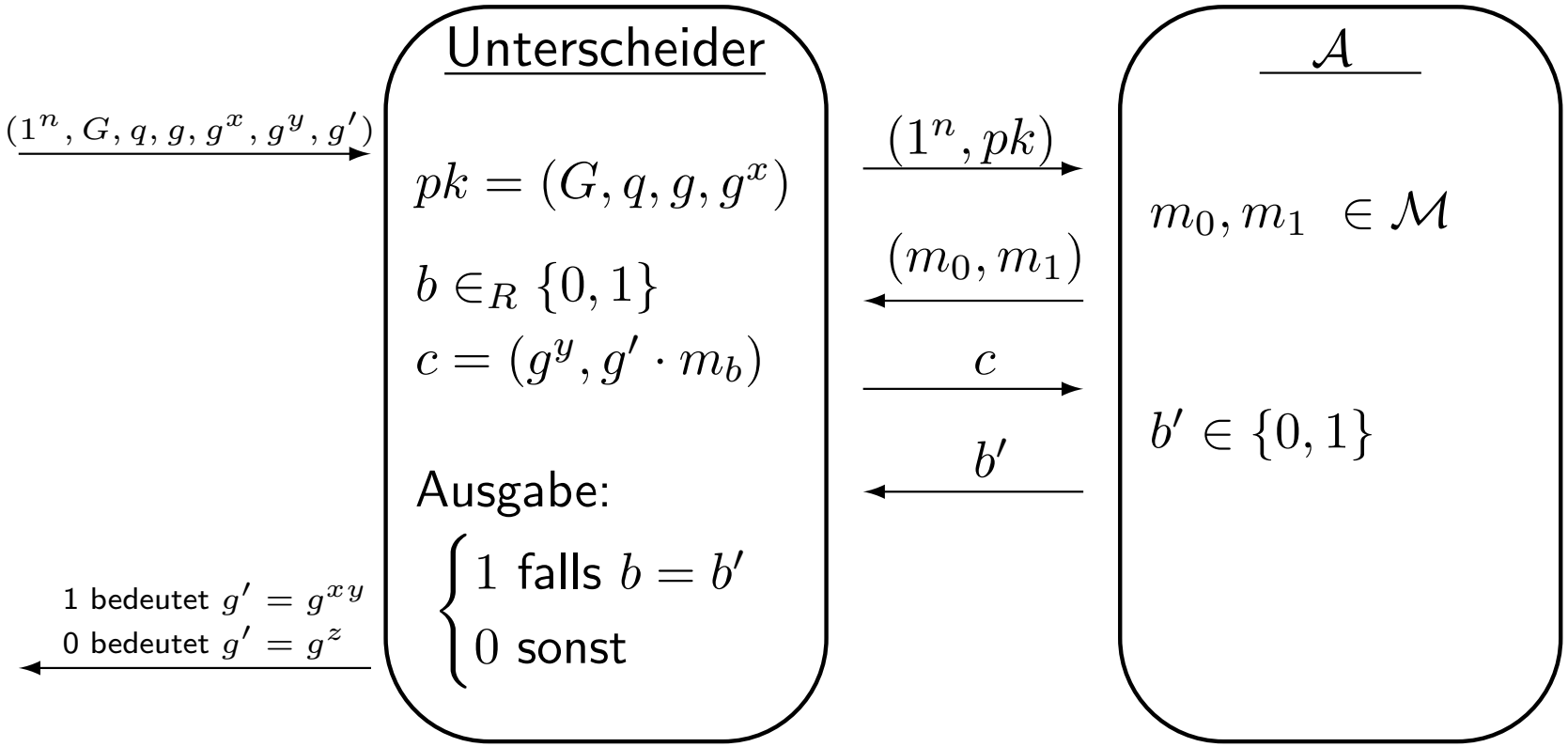
1. Setze $pk = (G, q, g, g^x)$.
2. $(m_0, m_1) \leftarrow \mathcal{A}(pk)$.
3. Wähle $b \in_R \{0, 1\}$ und berechne $b' \leftarrow \mathcal{A}(g^y, g' \cdot m_b)$.
4. Falls $b' = b$ Ausgabe 1, sonst Ausgabe 0.

AUSGABE: $\begin{cases} 1 & \text{wird interpretiert als } g' = g^{xy} \\ 0 & \text{wird interpretiert als } g' = g^z \end{cases}$.

Fall 1: Eingabe ist kein DDH-Tupel, d.h. $g' = g^z$ für $z \in_R \mathbb{Z}_q$.

- Chiffretext c ist wie bei Π' von der Form $(g^y, g^z \cdot m_b)$.
- Damit $\mathrm{Ws}[D(G, q, g, g^x, g^y, g^z) = 1] = \mathrm{Ws}[PubK_{\mathcal{A},\Pi'}(n) = 1] = \frac{1}{2}$.

DDH-Unterscheider mit Angreifer $\mathcal{A}$



Fall DDH-Tupel

Fall 2: Eingabe ist ein DDH-Tupel, d.h. $g' = g^{xy}$.

- $c = (g^y, g^{xy} \cdot m_b)$ ist identisch zu ElGamal-Chiffretexten verteilt.
- D.h. $\mathrm{Ws}[D(G, q, g, g^x, g^y, g^{xy}) = 1] = \mathrm{Ws}[PubK_{\mathcal{A},\Pi}(n) = 1] = \epsilon(n)$.
- Aus der DDH-Annahme folgt

$$\begin{aligned} \mathrm{negl}(n) &\geq |\mathrm{Ws}[D(G, q, g, g^x, g^y, g^z) = 1] - \mathrm{Ws}[D(G, q, g, g^x, g^y, g^{xy}) = 1]| \\ &= \left|\frac{1}{2} - \epsilon(n)\right|. \end{aligned}$$

- Daraus folgt $\epsilon(n) \leq \frac{1}{2} + \mathrm{negl}(n)$. □

Parameterwahl bei ElGamal

Einbetten von Nachrichten $\mathbf{m}' \in \{\mathbf{0},\mathbf{1}\}^*$

- Beliebte Parameterwahl: $\mathbb{Z}_p^*$, $p = 2q + 1$ mit p, q prim.
- D.h. p ist eine sogenannte starke Primzahl.
- **Ziel:** Untergruppe G mit primer Ordnung q.

- Quadrieren $\mathbb{Z}_p^* \to \mathbb{Z}_p^*$, $x \mapsto x^2$ ist eine $2-1$-Abbildung.
- Urbilder $x, p - x$ kollidieren, genau eines ist in $[\frac{p-1}{2}] = [q]$.
- Wir bezeichnen den Bildraum mit QR_p.
- QR_p ist Untergruppe von $\mathbb{Z}_p^*$ mit Ordnung q.
- Wählen g als Generator von QR_p. Sei $|q| = n$.
- Interpretieren $m' \in \{0,1\}^{n-1}$ als natürliche Zahl kleiner q.
- Es gilt $m' + 1 \in [q]$. Einbettung von m' ist $m = (m' + 1)^2 \bmod p$.
- Umkehren der Einbettung ist effizient berechenbar.

CPA-Sicherheit ist ungenügend

Definition CCA

CCA (=Chosen Ciphertext Attack) ist ein Angriff, bei dem der Angreifer sich Chiffretext seiner Wahl entschlüsseln lassen kann.

Beispiele in denen CPA nicht genügt:

- Eve fängt verschlüsselte Email $c = Enc(m)$ an Bob ab.
- Eve verschickt c selbst an Bob.
- Bob antwortet Eve und hängt dabei m an die Antwort an.
- D.h. Bob fungiert als Entschlüsselungsorakel.
- **Frage:** Lernt Eve Information, um andere c' zu entschlüsseln?

- Alice und Eve nehmen als Bieter an einer Auktion von Bob teil.
- Alice sendet ihr Gebot $c = Enc(m)$ verschlüsselt an Bob.
- *Enc* soll CPA-sicher sein, d.h. Eve erhält keine Information über m.
- **Frage:** Ist es Eve möglich, $c' = Enc(2m)$ aus c zu berechnen, ohne m zu kennen, und damit Alice zu überbieten? (Malleability)
- Man kann zeigen: CCA-sichere Verschlüsselung ist non-malleable

CCA Ununterscheidbarkeit

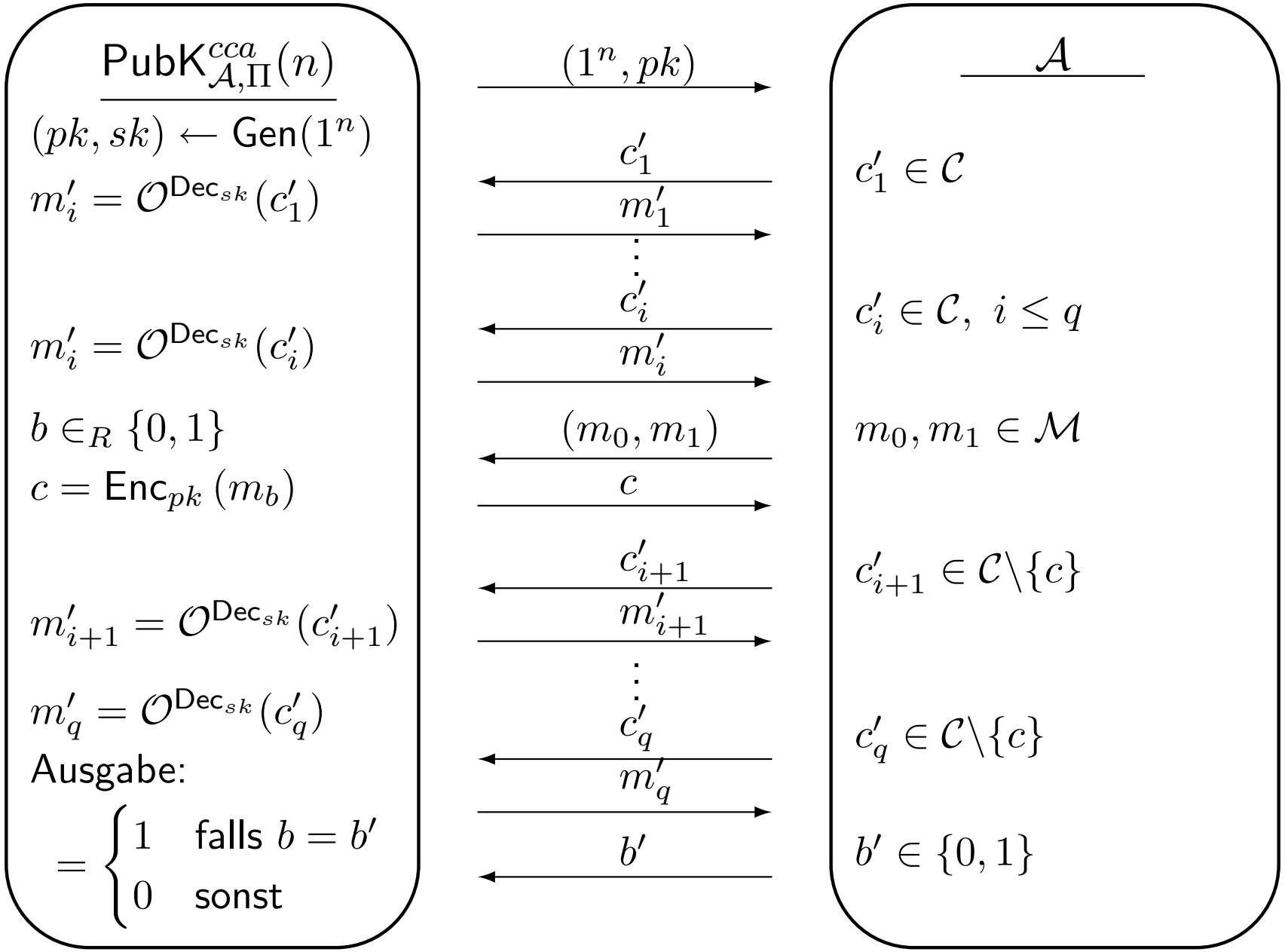
Spiel CCA Ununterscheidbarkeit von Chiffretexten $PubK^{cca}_{\mathcal{A},\Pi}(n)$

Sei Π ein PK-Verschlüsselungsverfahren mit Angreifer $\mathcal{A}$.

1. $(pk, sk) \leftarrow Gen(1^n)$
2. $(m_0, m_1) \leftarrow \mathcal{A}^{Dec_{sk}(\cdot)}(pk)$, wobei $Dec_{sk}(\cdot)$ ein Entschlüsselungsorakel für $\mathcal{A}$ für beliebige Chiffretexte ist.
3. Wähle $b \in_R \{0, 1\}$. Verschlüssele $c \leftarrow Enc_{pk}(m_b)$.
4. $b' \leftarrow \mathcal{A}^{Dec_{sk}(\cdot)}(c)$, wobei $\mathcal{A}$ beliebige Chiffretexte $c' \neq c$ durch das Orakel $Dec_{sk}(\cdot)$ entschlüsseln lassen darf.
5. $$PubK^{cca}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & \text{für } b = b' \\ 0 & \text{sonst} \end{cases}$$

Anmerkungen:

- Zusätzlich zum Verschlüsselungs-Orakel bei CPA besitzt $\mathcal{A}$ bei CCA ein weiteres Entschlüsselungs-Orakel $Dec_{sk}(\cdot)$.
- Falls $\mathcal{A}$ in Schritt 4 auch c entschlüsseln darf, ist das Spiel trivial.


$\mathsf{PubK}^{cca}_{\mathcal{A},\Pi}(n)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
$m'_i = \mathcal{O}^{\mathsf{Dec}_{sk}}(c'_1)$
$m'_i = \mathcal{O}^{\mathsf{Dec}_{sk}}(c'_i)$
$b \in_R \{0,1\}$
$c = \mathsf{Enc}_{pk}(m_b)$
$m'_{i+1} = \mathcal{O}^{\mathsf{Dec}_{sk}}(c'_{i+1})$
$m'_q = \mathcal{O}^{\mathsf{Dec}_{sk}}(c'_q)$
Ausgabe:
$= \begin{cases} 1 & \text{falls } b = b' \\ 0 & \text{sonst} \end{cases}$
$(1^n, pk)$
c'_1
m'_1
c'_i
m'_i
(m_0, m_1)
c
c'_{i+1}
m'_{i+1}
c'_q
m'_q
b'
$\mathcal{A}$
$c'_1 \in \mathcal{C}$
$c'_i \in \mathcal{C},\ i \leq q$
$m_0, m_1 \in \mathcal{M}$
$c'_{i+1} \in \mathcal{C} \backslash \{c\}$
$c'_q \in \mathcal{C} \backslash \{c\}$
$b' \in \{0,1\}$

CCA-Sicherheit

Definition CCA-Sicherheit

Ein Verschlüsselungsverfahren Π heißt *CCA-sicher* bzw. besitzt *ununterscheidbare Chiffretexte unter CCA*, falls für alle ppt Angreifer $\mathcal{A}$ gilt $\mathrm{Ws}[PubK^{cca}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathrm{negl}(n)$.

Anmerkungen:

- Erstes effizientes CCA-sicheres Verfahren Cramer-Shoup (1998).
- Cramer-Shoup verwendet die DDH-Annahme.
- Chiffretexte sind doppelt so lang wie bei ElGamal.
- Sicherheitsbeweis von Cramer-Shoup ist nicht-trivial.
- Später in der Vorlesung: CCA-sichere Verfahren im sogenannten Random Oracle Modell.

CCA Angriff und Malleability von Textbuch RSA

CCA Angriff auf Textbuch RSA

- Wollen $c = m^e \bmod N$ entschlüsseln.
- Man beachte: c darf nicht direkt angefragt werden.
- Berechne $c' = c \cdot r^e = (mr)^e \bmod N$ für $r \in \mathbb{Z}_N^* \setminus \{1\}$.
- Berechne $mr \leftarrow Dec_{sk}(c')$ mittels Entschlüsselungs-Orakel.
- Berechne $mr \cdot r^{-1} = m \bmod N$.

Malleability von Textbuch RSA

- Voriger Angriff zeigt: Für $c = m^e \bmod N$ kann die Verschlüsselung von mr berechnet werden, ohne m selbst zu kennen.
- D.h. Textbuch RSA ist malleable.

CCA Angriff und Malleability von ElGamal

Praktischer CCA-Angriff auf Padded RSA Variante PKCS #1 v1.5

- Bleichenbacher Angriff: Sende adaptiv Chiffretexte an Server.
- Falls die Entschlüsselung nicht das korrekte Format besitzt, sendet der Server eine Fehlermeldung zurück.
- Genügt, um einen beliebigen Chiffretext c zu entschlüsseln.

CCA Angriff auf ElGamal

- Ziel: Entschlüssele $c = (g^y, g^{xy} \cdot m)$.
- Lasse $c' = (g^y, g^{xy} \cdot m \cdot r)$ für $r \in G \setminus \{1\}$ entschlüsseln.
- Berechne $mr \cdot r^{-1} = m$.
- ElGamal ist malleable, da c' korrekte Verschlüsselung von mr.

Einwegfunktionen

Ziel: CPA-sichere Verschlüsselung aus Trapdoor-Einwegpermutation

Später: CCA-sichere Verschlüsselung aus Trapdoor-Einwegperm.

Spiel Invertieren $Invert_{\mathcal{A},f}(n)$

Sei $f : \{0,1\}^* \to \{0,1\}^*$ effizient berechenbar, $\mathcal{A}$ ein Invertierer für f.

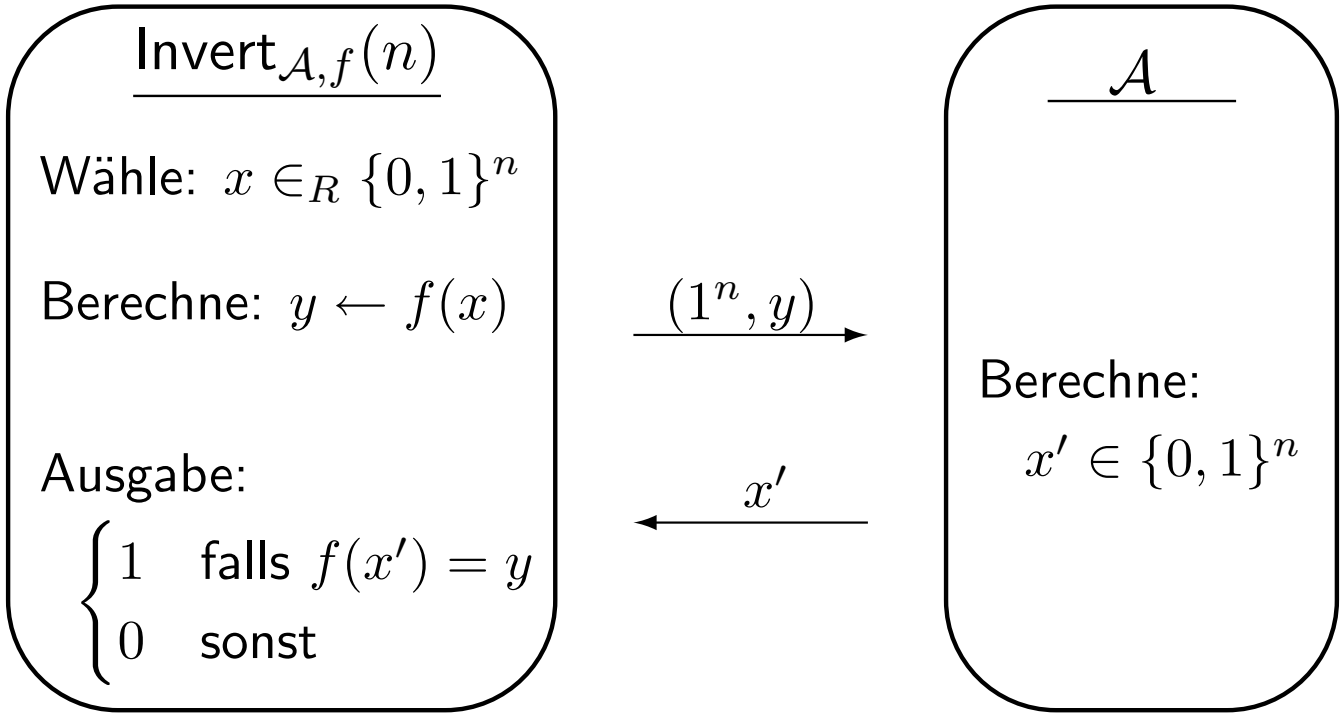
1. Wähle $x \in_R \{0,1\}^n$. Berechne $y \leftarrow f(x)$.
2. $x' \leftarrow \mathcal{A}(1^n, y)$
3. $Invert_{\mathcal{A},f}(n) = \begin{cases} 1 & \text{falls } f(x') = y \\ 0 & \text{sonst} \end{cases}$.

Definition Einwegfunktion

Eine Funktion $f : \{0,1\}^* \to \{0,1\}^*$ heißt *Einwegfunktion*, falls

1. Es existiert ein deterministischer pt Alg $\mathcal{B}$ mit $f(x) \leftarrow \mathcal{B}(x)$.
2. Für alle ppt Algorithmen $\mathcal{A}$ gilt $\mathrm{Ws}[Invert_{\mathcal{A},f}(n) = 1] \leq \mathrm{negl}(n)$.

Spiel Invertieren



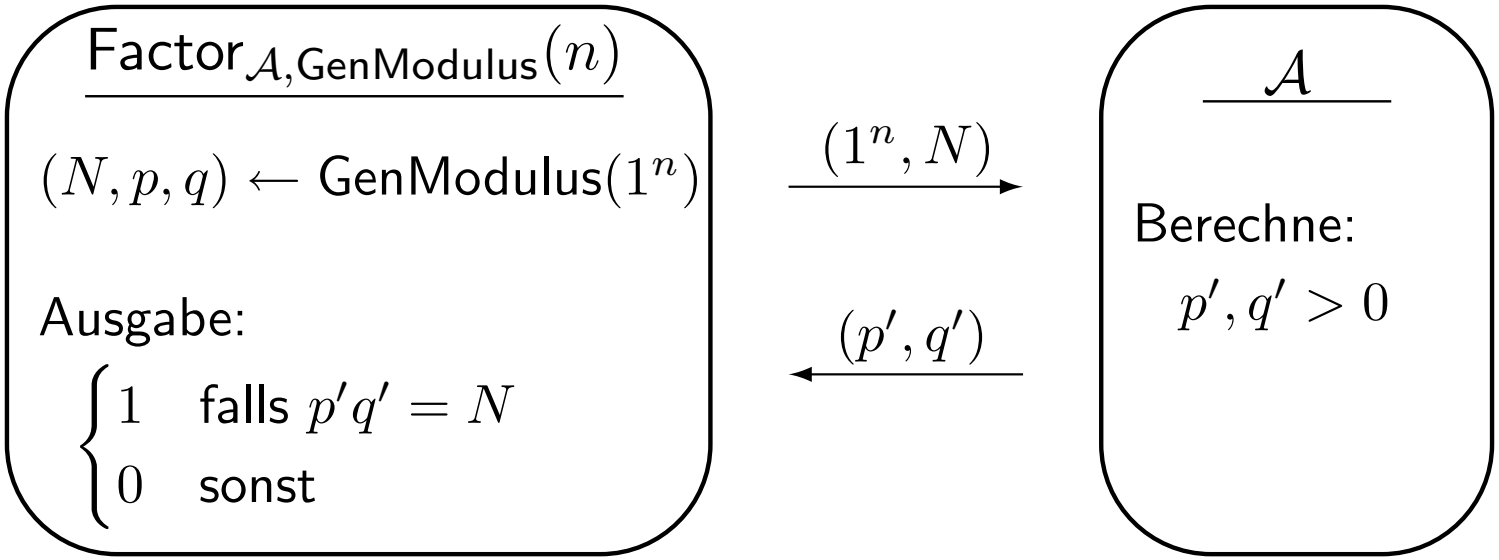
Die Faktorisierungsannahme

- **Problem:** Existenz von Einwegfunktionen ist ein offenes Problem.
- Konstruktion unter Komplexitätsannahme (z.B. Faktorisierung)
- Verwenden dazu $(N, p, q) \leftarrow \mathit{GenModulus}(1^n)$ von RSA.

Spiel Faktorisierungsspiel $\mathit{Factor}_{\mathcal{A},\mathit{GenModulus}}(n)$

1. $(N, p, q) \leftarrow \mathit{GenModulus}(1^n)$
2. $(p', q') \leftarrow \mathcal{A}(N)$ mit $p', q' > 1$.
3. $\mathit{Factor}_{\mathcal{A},\mathit{GenModulus}}(n) = \begin{cases} 1 & \text{falls } p'q' = N \\ 0 & \text{sonst} \end{cases}$.

Spiel Faktorisieren



Definition Faktorisierungsannahme

Faktorisieren ist hart bezüglich *GenModulus* falls für alle ppt Algorithmen $\mathcal{A}$ gilt $\mathrm{Ws}[\mathit{Factor}_{\mathcal{A},\mathit{GenModulus}}(n) = 1] \leq \mathrm{negl}(n)$.

Faktorisierungsannahme: Faktorisieren ist hart bezüglich *GenModulus*.

Konstruktion aus Faktorisierungsannahme

- Sei $p(n)$ ein Polynom, so dass $\mathit{GenModulus}(1^n)$ höchstens $p(n)$ Zufallsbits verwendet.
- OBdA sei $p(n) : \mathbb{N} \to \mathbb{N}$ monoton wachsend.

Algorithmus FACTOR-ONEWAY f_{FO}

Eingabe: $x \in \{0,1\}^*$

1. Berechne n mit $p(n) \leq |x| < p(n+1)$.
2. $(N, p, q) \leftarrow \mathit{GenModulus}(1^n, x)$, wobei *GenModulus* die Eingabe x als internen Zufallsstring verwendet.

Ausgabe: N

Bemerkung:

- $\mathit{GenModulus}(1^n, x)$ ist deterministisch. (Derandomisierung)