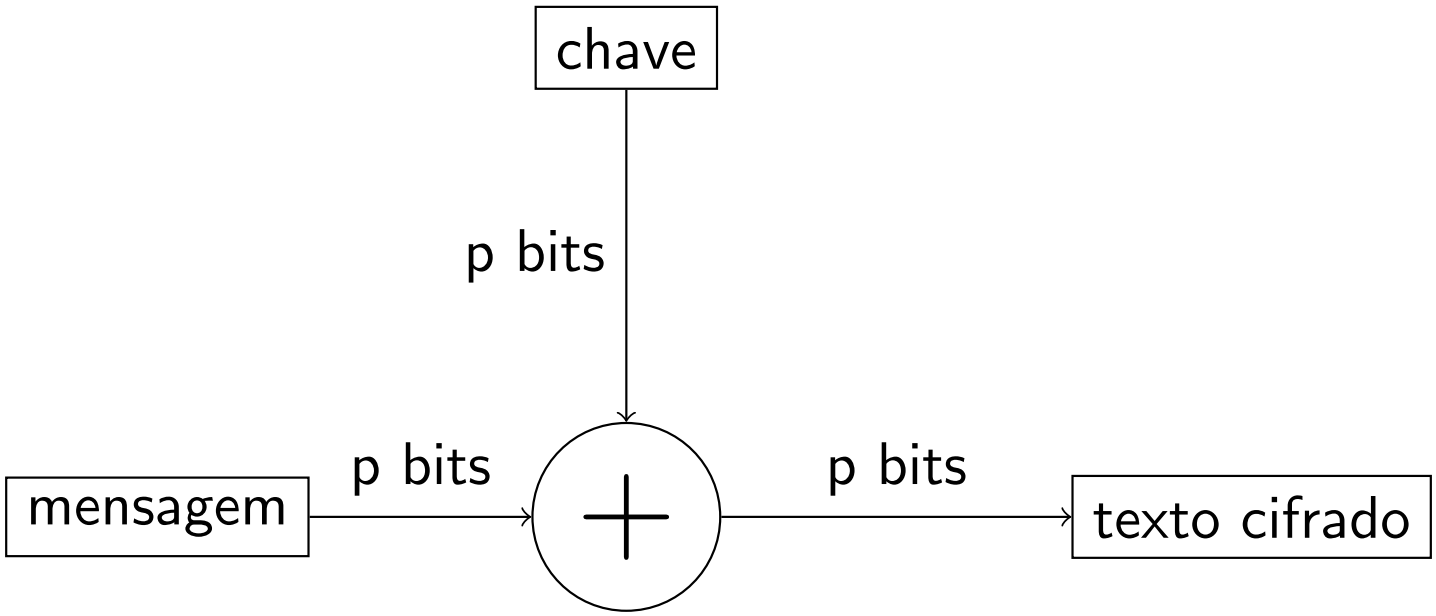
GBC083 - Segurança da Informação
Aula 3 - Pseudo-aleatoriedade

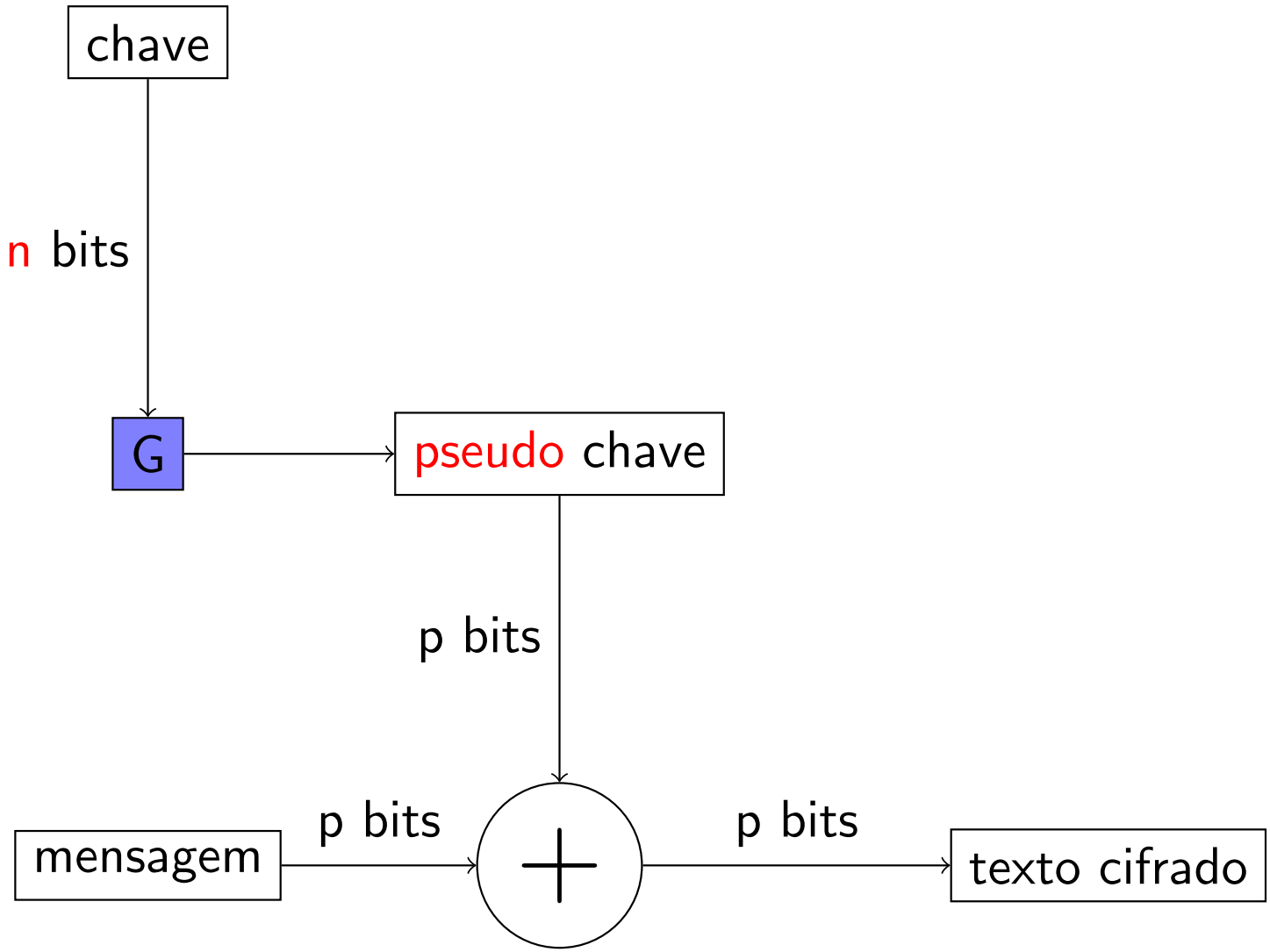
Pseudo One-Time Pad - resumo até agora

- Vimos que existem restrições para sigilo perfeito
  - Ineficiência: chave do tamanho da mensagem
- Definimos sigilo computacional: noção mais prática de segurança
- Como superar limitações do One-Time Pad - Cifra de Uso Único

Revisão: one-time pad - cifra de uso único



Pseudo one-time pad



Pseudo one-time pad

- Seja G função polinomial determinística com $|G(k)| = p(|k|)$
- $Gen(1^n)$: produz chave uniforme k de n bits
  - Parâmetro de segurança $n \Rightarrow$ espaço de mensagem $\{0,1\}^{p(n)}$
- $Enc_k(m)$: produz $G(k) \oplus m$
- $Dec_k(c)$: produz $G(k) \oplus c$

Segurança do pseudo one-time pad

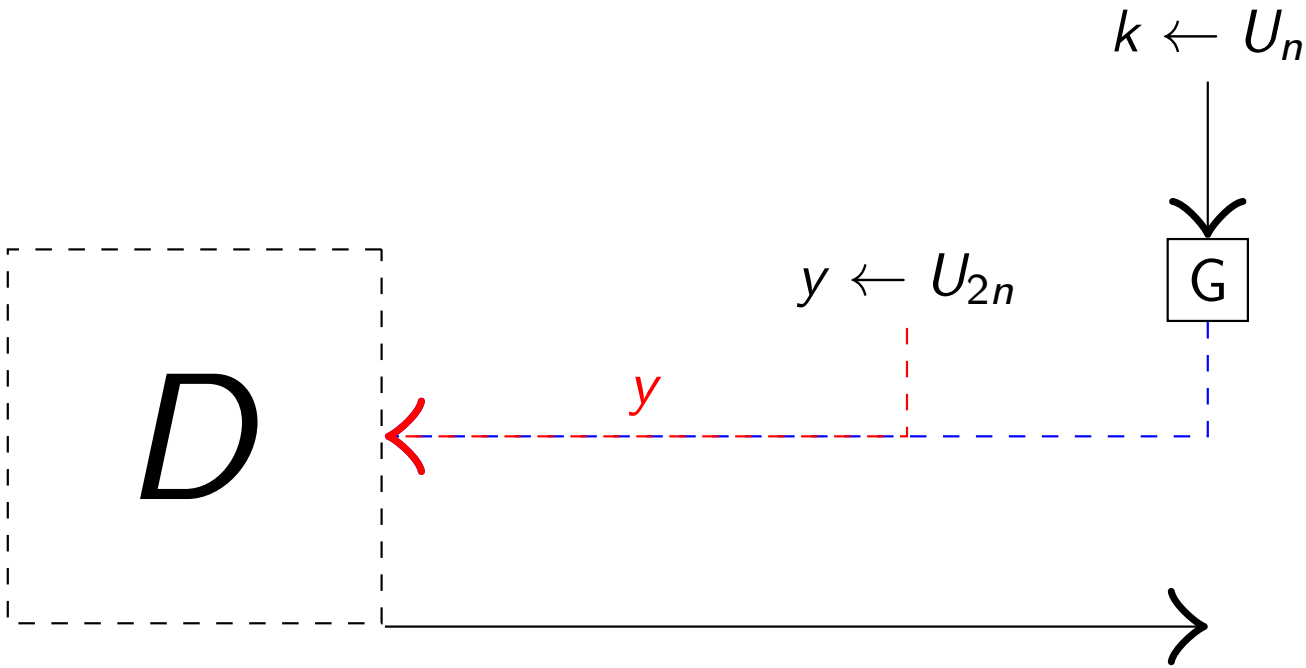
- Provar segurança do pseudo one-time pad sabendo que G é um GPA
  - Prova por redução

Segurança do OTP com um pseudo-GPA

- Se G é um gerador pseudo-aleatório, então o pseudo-OTP é computacionalmente indiscernível
- Meta é provar que pseudo one-time pad atende a essa definição
- Não é possível provar isso de maneira incondicional
  - Além das técnicas atuais
  - Segurança depende de G
- Possível provar segurança assumindo que G é um GPA

Recapitulação de GPA

- Seja G uma função eficiente determinística com $|G(k)| = 2 \cdot |k|$

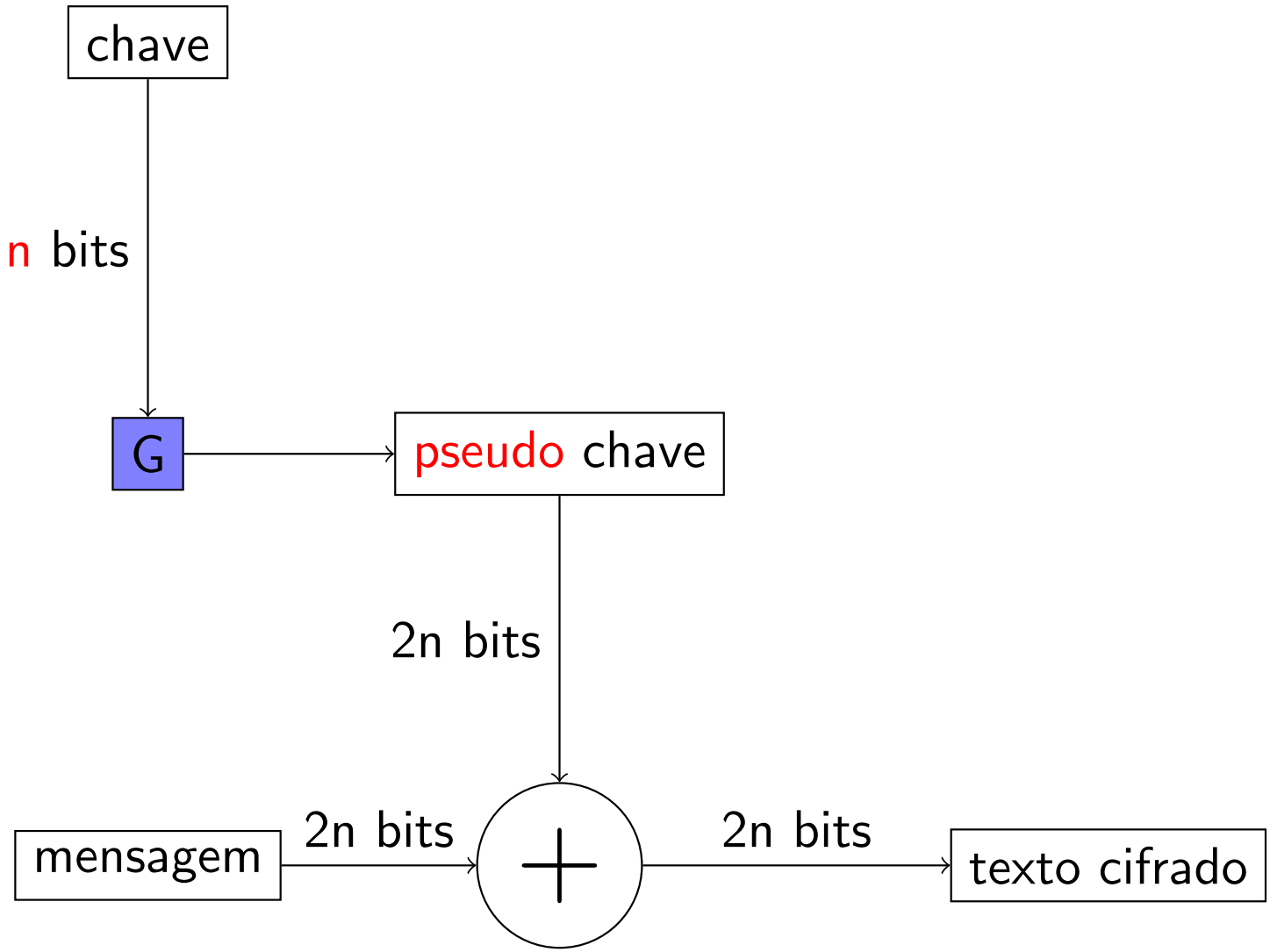


- Para qualquer atacante D eficiente, as probabilidades de que D produz 1 (ou seja, julga y como aleatório) em ambos casos (y vir de U_{2n} ou de $G(\cdot)$) devem ser próximas

Ideia da prova

- Assuma que G é um gerador pseudo-aleatório
- Fixar um atacante eficiente arbitrário A atacando o esquema pseudo-OTP
- Usar A como subrotina para construir um D eficiente atacando G
  - Relacionar a probabilidade de discernibilidade de D à probabilidade de sucesso de A
- Por hipótese, a probabilidade de sucesso de D deve ser negligível
  - Limita a probabilidade de sucesso de A

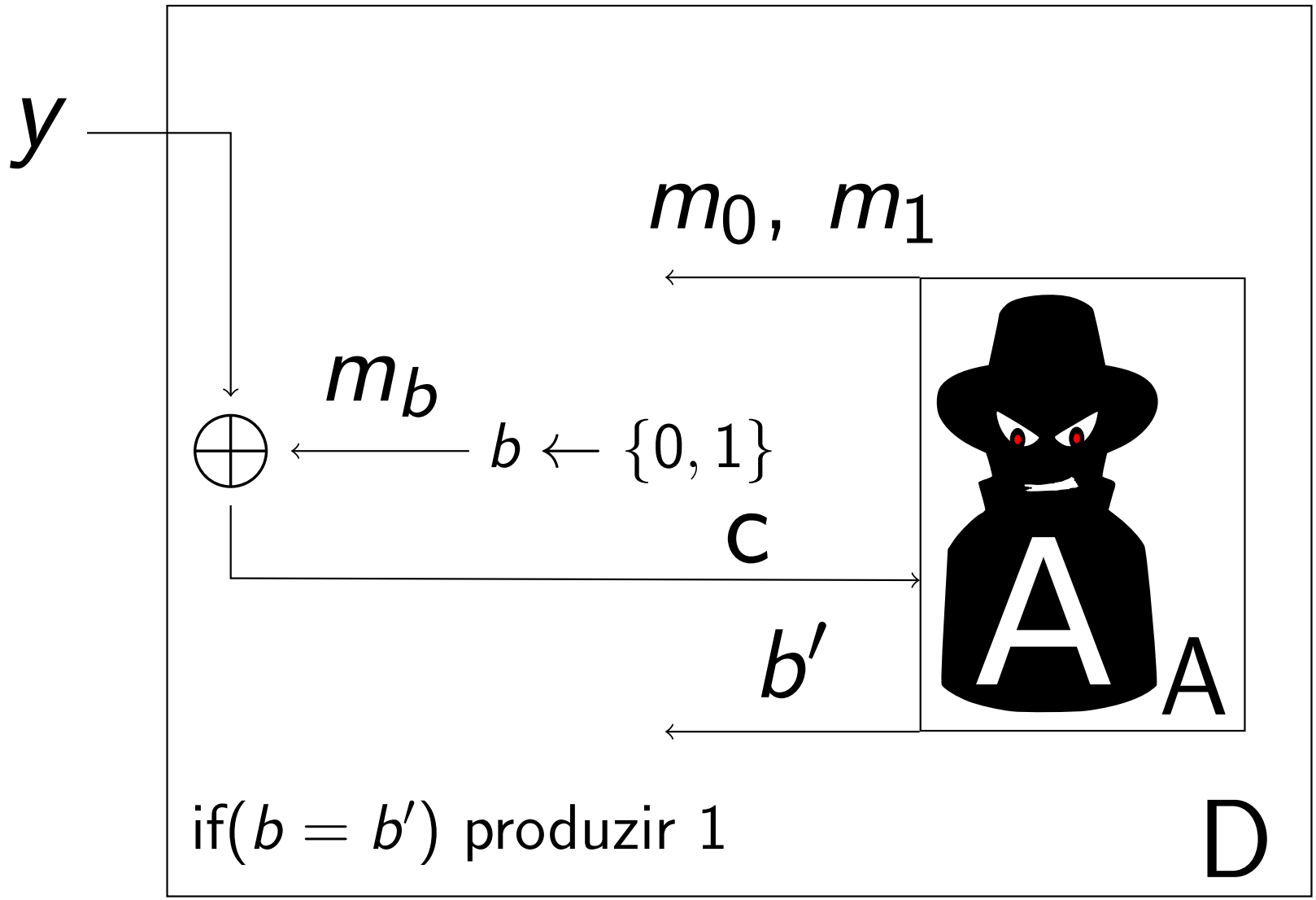
Pseudo one-time pad



Teorema da Segurança

- Se G é um gerador pseudo-aleatório, então o pseudo one-time pad é computacionalmente indiscernível

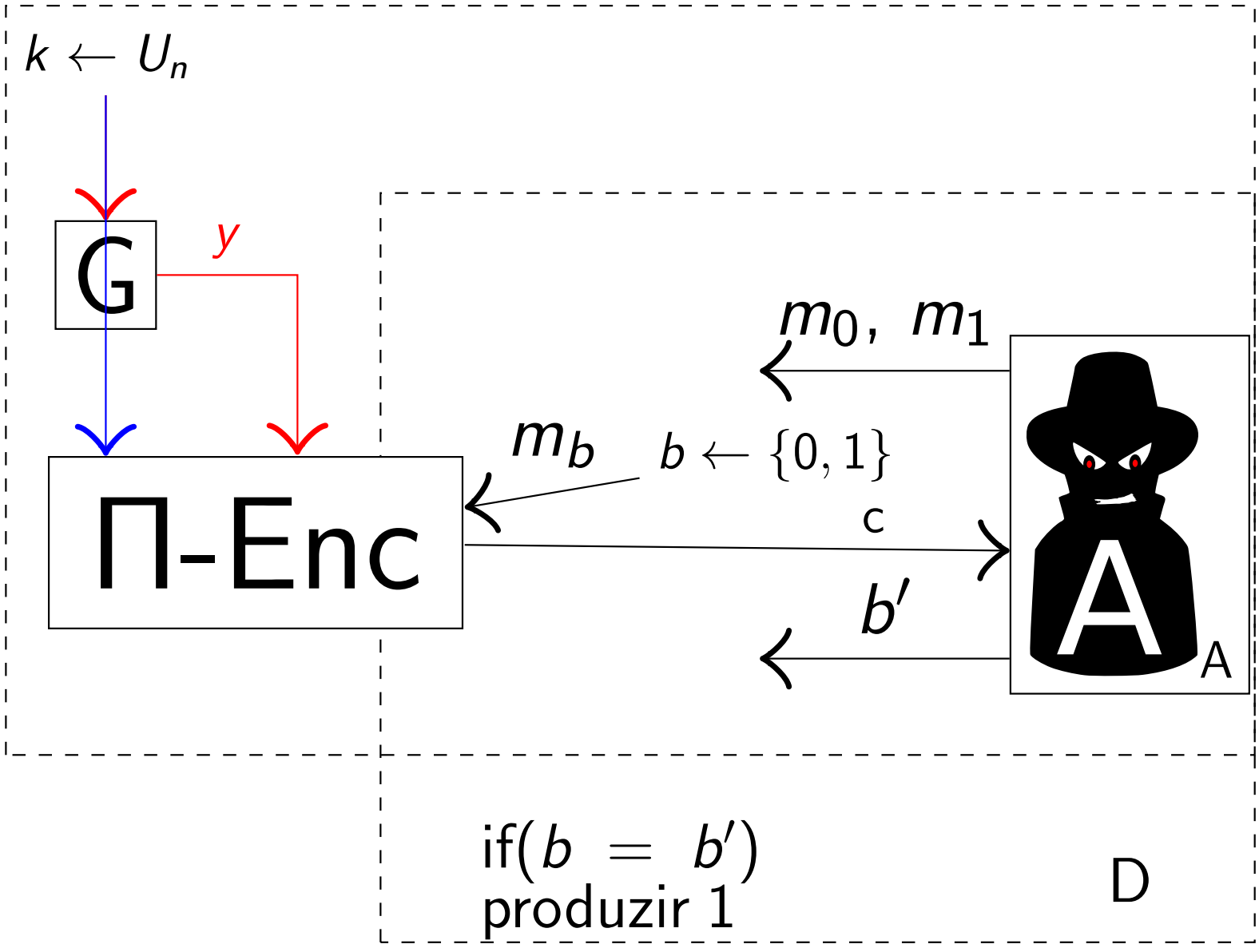
A redução: D vai usar atacante A



Análise

- Se A roda em tempo polinomial, então D também
- Seja $\mu(n) = P(PrivK_{A,\Pi}(n) = 1)$
- Se entrada y é pseudo-aleatória, a visão de A é exatamente como em $PrivK_{A,\Pi}(n)$
  - $\Rightarrow P_{x \leftarrow U_n}(D(G(x)) = 1) = \mu(n)$

A redução


$k \leftarrow U_n$
G
y
$m_0,\ m_1$
m_b
$b \leftarrow \{0,1\}$
Π-Enc
c
b'
A
A
if($b = b'$)
produzir 1
D

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5
  - $\Rightarrow P_{x \leftarrow U_{2n}}(D(y) = 1) = 0.5$

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5
  - $\Rightarrow P_{x \leftarrow U_{2n}}(D(y) = 1) = 0.5$
- Se entrada y é pseudo-aleatória, a visão de A é exatamente como em $PrivK_{A,\Pi}$

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5
  - $\Rightarrow P_{x \leftarrow U_{2n}}(D(y) = 1) = 0.5$
- Se entrada y é pseudo-aleatória, a visão de A é exatamente como em $PrivK_{A,\Pi}$
  - Seja $\mu(n) = P(PrivK_{A,\Pi}(n) = 1)$

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5
  - $\Rightarrow P_{x \leftarrow U_{2n}}(D(y) = 1) = 0.5$
- Se entrada y é pseudo-aleatória, a visão de A é exatamente como em $PrivK_{A,\Pi}$
  - Seja $\mu(n) = P(PrivK_{A,\Pi}(n) = 1)$
  - $\Rightarrow P_{x \leftarrow U_n}(D(G(x)) = 1) = \mu(n)$

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5
  - $\Rightarrow P_{x \leftarrow U_{2n}}(D(y) = 1) = 0.5$
- Se entrada y é pseudo-aleatória, a visão de A é exatamente como em $PrivK_{A,\Pi}$
  - Seja $\mu(n) = P(PrivK_{A,\Pi}(n) = 1)$
  - $\Rightarrow P_{x \leftarrow U_n}(D(G(x)) = 1) = \mu(n)$
- Como G é pseudoaleatório...

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5
  - $\Rightarrow P_{x \leftarrow U_{2n}}(D(y) = 1) = 0.5$
- Se entrada y é pseudo-aleatória, a visão de A é exatamente como em $PrivK_{A,\Pi}$
  - Seja $\mu(n) = P(PrivK_{A,\Pi}(n) = 1)$
  - $\Rightarrow P_{x \leftarrow U_n}(D(G(x)) = 1) = \mu(n)$
- Como G é pseudoaleatório...
- Use seja, vale

$$|P_{x \leftarrow U_n}(A(G(x)) = 1) - P_{y \leftarrow U_{2n}}(A(y) = 1)| \leq \epsilon(n)$$

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5
  - $\Rightarrow P_{x \leftarrow U_{2n}}(D(y) = 1) = 0.5$
- Se entrada y é pseudo-aleatória, a visão de A é exatamente como em $PrivK_{A,\Pi}$
  - Seja $\mu(n) = P(PrivK_{A,\Pi}(n) = 1)$
  - $\Rightarrow P_{x \leftarrow U_n}(D(G(x)) = 1) = \mu(n)$
- Como G é pseudoaleatório...
- Use seja, vale

$$|P_{x \leftarrow U_n}(A(G(x)) = 1) - P_{y \leftarrow U_{2n}}(A(y) = 1)| \leq \epsilon(n)$$

- $|\mu(n) - 0.5| \leq \epsilon(n)$

Análise

- Queremos provar que pseudo-OTP é computacionalmente indiscernível
- Se entrada y é uniforme, A é bem sucedido com probabilidade 0.5
  - $\Rightarrow P_{x \leftarrow U_{2n}}(D(y) = 1) = 0.5$
- Se entrada y é pseudo-aleatória, a visão de A é exatamente como em $PrivK_{A,\Pi}$
  - Seja $\mu(n) = P(PrivK_{A,\Pi}(n) = 1)$
  - $\Rightarrow P_{x \leftarrow U_n}(D(G(x)) = 1) = \mu(n)$
- Como G é pseudoaleatório...
- Use seja, vale

$$|P_{x \leftarrow U_n}(A(G(x)) = 1) - P_{y \leftarrow U_{2n}}(A(y) = 1)| \leq \epsilon(n)$$

  - $|\mu(n) - 0.5| \leq \epsilon(n)$
  - $\Rightarrow P(PrivK_{A,\Pi}(n) = 1) \leq 0.5 + \epsilon(n)$

O que isso tudo significa?

- Prova que o pseudo OTP é seguro
  - Temos um esquema provado como seguro em vez de uma construção heurística
- Com alguns poréns:
  - assumindo G é um gerador pseudo-aleatório
- A única forma que o esquema pode ser quebrado é:
  - Se uma fraqueza é achada em G
  - Se a definição de pseudo-aleatoriedade não é forte o suficiente