Introduction to Modern Cryptography

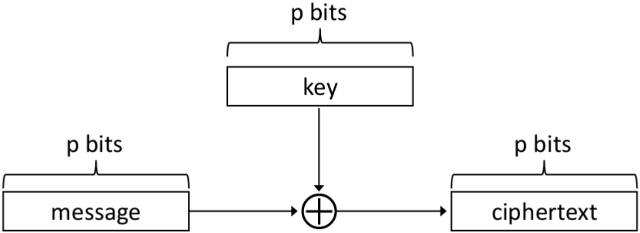
Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

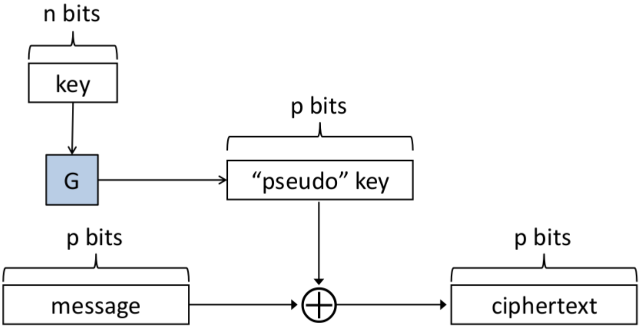
Lecture 6

Pseudo One-Time Pad

One-time Pad (recall)



Pseudo One-time Pad (POTP)



Pseudo One-time Pad

Definition

- Let G be a deterministic algorithm, with $|G(k)| = p(|k|)$
- $\mathsf{Gen}(1^n)$: output uniform n-bit key k
  - Security parameter $n \implies$ message space $\{0,1\}^{p(n)}$
- $\mathsf{Enc}_k(m)$: output $G(k) \oplus m$
- $\mathsf{Dec}_k(c)$: output $G(k) \oplus c$
- Correctness – the same as OTP

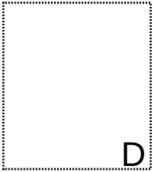
Security of POTP?

- Would like to be able to prove security
- Based on the assumption that $\boldsymbol{G}$ is a PRG

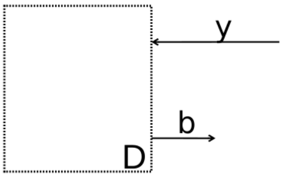
Modern Crypto = Definitions + Proofs + Assumptions

- We've **defined** computational secrecy
- Our goal is to **prove** that the pseudo OTP meets that definition
- We cannot prove this unconditionally
  - Beyond our current techniques...
  - Anyway, security clearly depends on $\boldsymbol{G}$
- Can prove security based on the **assumption** that $\boldsymbol{G}$ is a pseudorandom generator

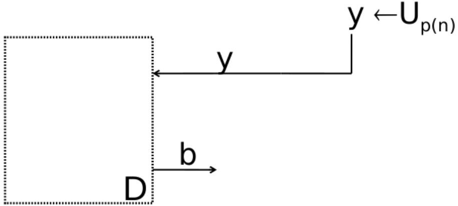
PRG Revisited



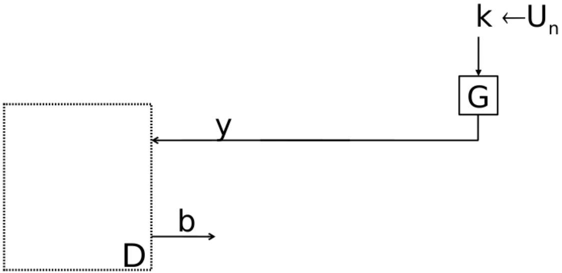
PRG Revisited



PRG Revisited

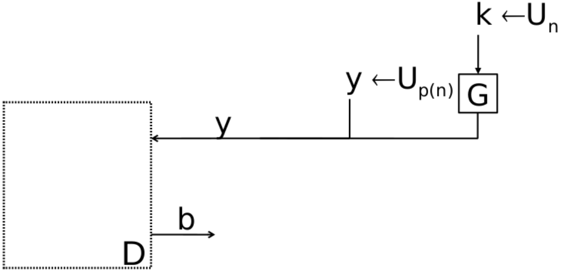


PRG Revisited



PRG Revisited

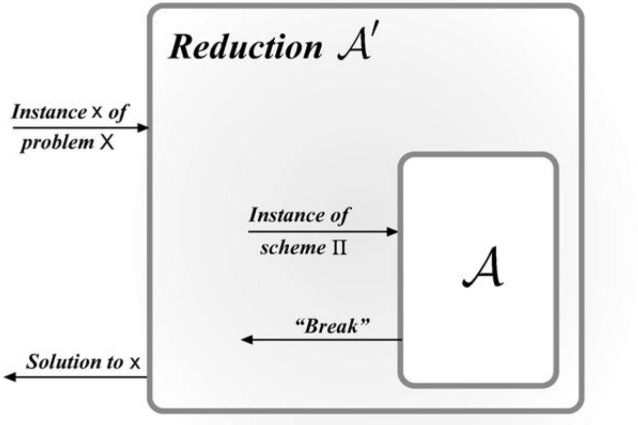
- Let G be an efficient, deterministic function with $|G(k)| = p(|k|)$
- For any efficient D, the probabilities that D outputs $\mathbf{1}$ in each case must be *close*



Proof by Reduction

- Assume $\boldsymbol{G}$ is a pseudorandom generator
- Assume toward a contradiction that there is an efficient attacker $\boldsymbol{A}$ who *breaks* POTP (as per the definition)
- Use $\boldsymbol{A}$ as a subroutine to build an efficient $\boldsymbol{D}$ that *breaks* pseudorandomness of $\boldsymbol{G}$
- By assumption, no such $\boldsymbol{D}$ exists
- $\implies$ No such $\boldsymbol{A}$ can exist

Proof by Reduction



IMC Textbook 2nd ed. CRC Press 2015

Proof by Reduction (equivalent)

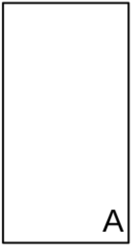
- Assume G is a pseudorandom generator
- Fix some arbitrary, efficient A attacking POTP
- Use A as a subroutine to build an efficient D attacking G
- **Relate the distinguishing gap of D to the success probability of A**
- By assumption, the distinguishing gap of D must be negligible
- $\implies$ Use this to bound the success probability of A

Security of POTP

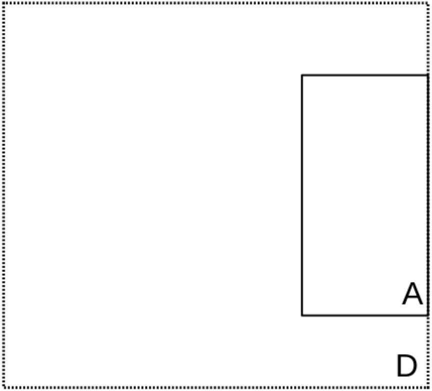
Theorem

If $\boldsymbol{G}$ is a pseudorandom generator, then the pseudo one-time pad $\mathbf{\Pi}$ is EAV-secure (i.e. computationally indistinguishable)

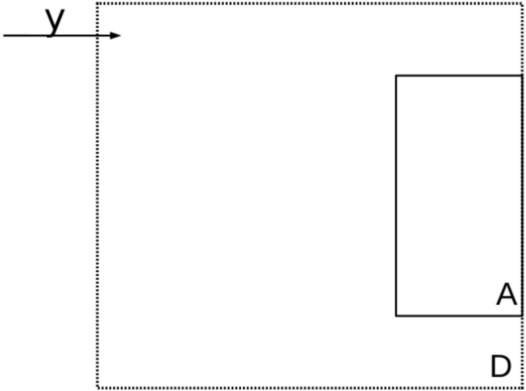
The Reduction



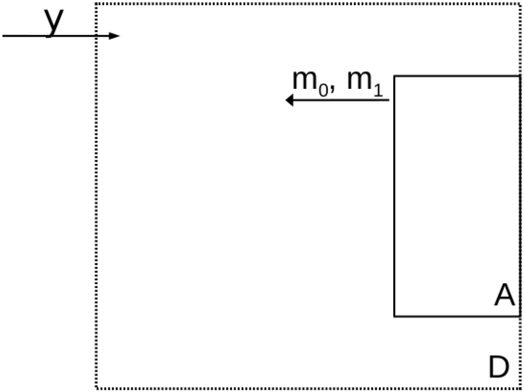
The Reduction



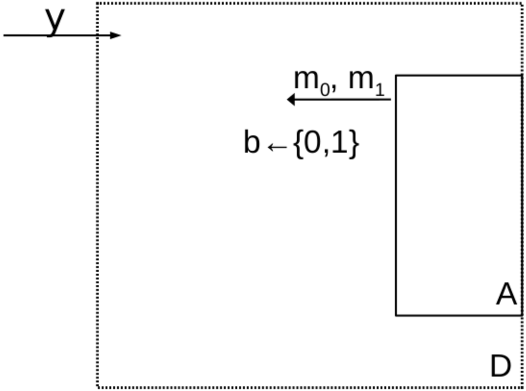
The Reduction



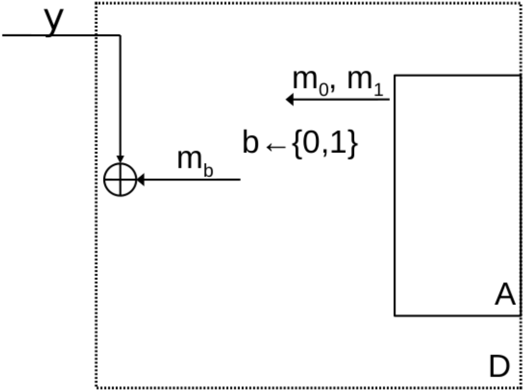
The Reduction



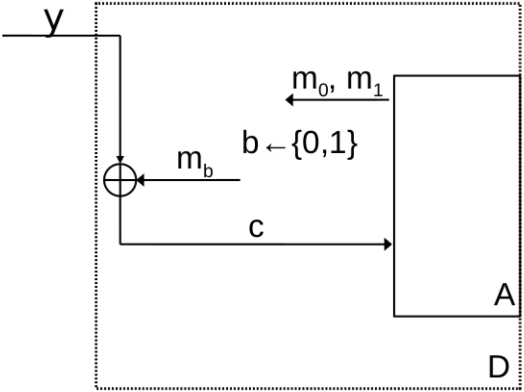
The Reduction



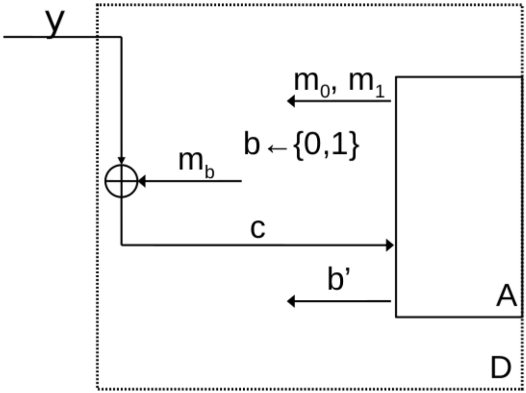
The Reduction



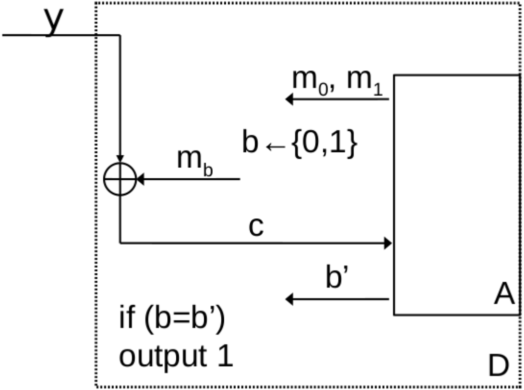
The Reduction



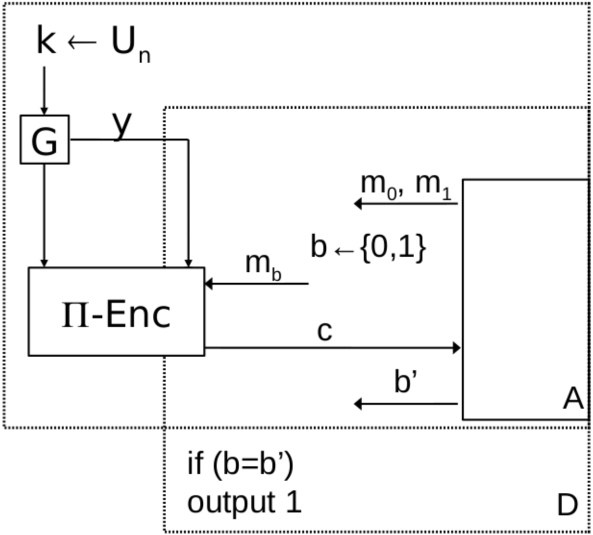
The Reduction



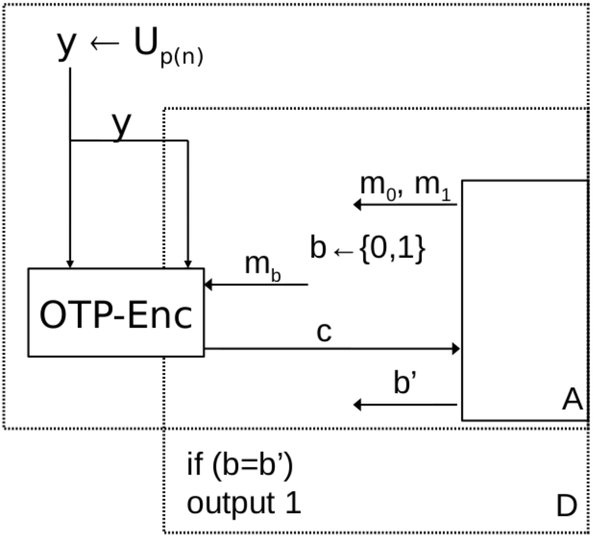
The Reduction



The Reduction



The Reduction



The Proof

Proof by Reduction

- Implement $\boldsymbol{D}$ by using $\boldsymbol{A}$ as a subroutine
  - If $\boldsymbol{A}$ runs in polynomial time, then so does $\boldsymbol{D}$
- Relate the success $\mathbf{Pr}$ of $\boldsymbol{D}$ and $\boldsymbol{A}$
- Prove that if $\boldsymbol{A}$ succeeds in breaking POTP then $\boldsymbol{D}$ succeeds in breaking $\boldsymbol{G}$
- i.e. **reduce the security of the POTP to the security of the underlying $\boldsymbol{G}$**

The Attacker A

A attacks POTP via $\mathsf{PrivK}_{A,\Pi}(n)$

- $A(1^n)$ outputs m_0, m_1
- $k \leftarrow \mathsf{Gen}(1^n)$, $b \leftarrow \{0, 1\}$, $c \leftarrow \mathsf{Enc}_k(m_b)$
- $b' \leftarrow A(c)$
- If $b = b'$ return 1 (success)

If POTP is computationally ind. (EAV-secure) then

$$\mathbf{Pr}[\mathsf{PrivK}_{A,\Pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$

$\implies$ sufficient to prove the above inequality in order to prove the security of the POTP

The Attacker A

A attacks OTP via $\mathsf{PrivK}_{A,\mathsf{OTP}}$

1. A outputs m_0, m_1
2. $k \leftarrow \mathsf{Gen}$, $b \leftarrow \{0, 1\}$, $c \leftarrow \mathsf{Enc}_k(m_b)$
3. $b' \leftarrow A(c)$
4. If $b = b'$ return 1 (success)

Since OTP is perfectly secret:

$$\mathbf{Pr}[\mathsf{PrivK}_{A,\mathsf{OTP}} = 1] = \frac{1}{2}$$

The Distinguisher $\boldsymbol{D}$

$\boldsymbol{D}$ attacks $\boldsymbol{G}$

Since $\boldsymbol{G}$ is a PRG (by assumption) $\implies \exists \boldsymbol{\epsilon(n) = \mathbf{negl}}$ s.t.

$$\boldsymbol{|\mathbf{Pr}_{x \leftarrow U_n}[D(G(x)) = 1] - \mathbf{Pr}_{y \leftarrow U_{p(n)}}[D(y) = 1]| \leq \epsilon(n)}$$

World $\mathbf{0}$: D with a Truly Random Input

$D(y)$ for uniform y

D simulates the $\mathsf{PrivK}_{A,\mathsf{OTP}}$ experiment for A for a truly random input y:

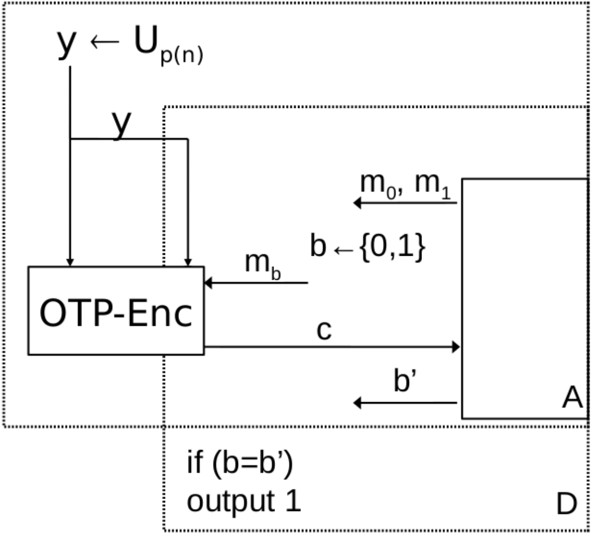
- $A(1^n)$ outputs m_0, m_1
- Simulation:
  1. D generates $b \leftarrow \{0, 1\}$
  2. D computes $c = m_b \oplus y$
  3. D sends c to A
- $b' \leftarrow A(c)$
- If $b = b'$ then $D(y) = 1$

World $\mathbf{0}$: $\boldsymbol{D}$ with a Truly Random Input

Since $\boldsymbol{y}$ is truly random, from the viewpoint of $\boldsymbol{A}$ it is as if $\boldsymbol{A}$ is interacting with the OTP in World $\mathbf{0}$. Therefore:

$$\mathbf{Pr}_{\boldsymbol{y} \leftarrow \boldsymbol{U}_{\boldsymbol{p(n)}}}[\boldsymbol{D(y)} = \mathbf{1}] = \mathbf{Pr}[\mathsf{PrivK}_{\boldsymbol{A},\mathsf{\mathbf{OTP}}} = \mathbf{1}] = \frac{\mathbf{1}}{\mathbf{2}}$$

World **0**: ***A*** interacting with OTP



World $\mathbf{1}$: D with a Pseudorandom Input

$D(G(x))$ for pseudorandom $G(x)$

D simulates the $\mathsf{PrivK}_{A,\Pi}(n)$ experiment for A for a pseudorandom input $G(x)$:

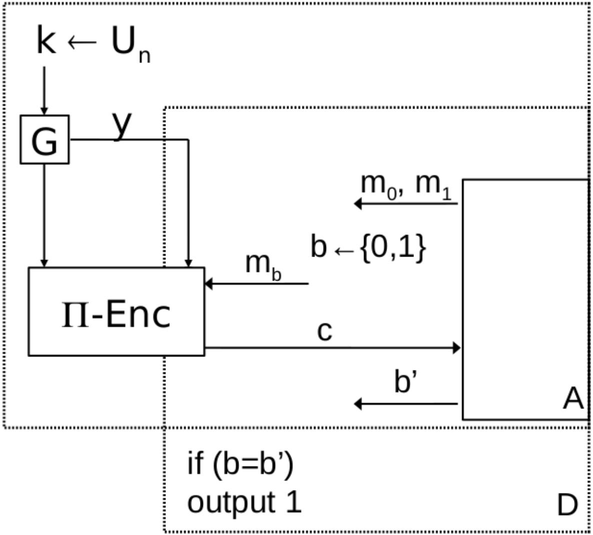
- $A(1^n)$ outputs m_0, m_1
- Simulation:
  1. D generates $b \leftarrow \{0,1\}$
  2. D computes $c = m_b \oplus G(x)$
  3. D sends c to A
- $b' \leftarrow A(c)$
- If $b = b'$ then $D(G(x)) = 1$

World $\mathbf{1}$: $\boldsymbol{D}$ with a Pseudorandom Input

Since $\boldsymbol{G(x)}$ is pseudorandom, from the viewpoint of $\boldsymbol{A}$ it is as if $\boldsymbol{A}$ is interacting with the POTP in World $\mathbf{1}$. Therefore:

$$\mathbf{Pr}_{x \leftarrow U_n}[D(G(x)) = 1] = \mathbf{Pr}[\mathsf{PrivK}_{A,\Pi}(n) = 1]$$

World **1**: ***A*** interacting with POTP



Proof.

1) By the assumption that G is a PRG $\exists \epsilon(n) = \mathbf{negl}$:

$$\left|\mathbf{Pr}_{x \leftarrow U_n}[D(G(x)) = 1] - \mathbf{Pr}_{y \leftarrow U_{p(n)}}[D(y) = 1]\right| \leq \epsilon(n)$$

2) By the simulation of $\mathsf{PrivK}_{A,\Pi}$ by $D(y)$:

$$\mathbf{Pr}_{y \leftarrow U_{p(n)}}[D(y) = 1] = \mathbf{Pr}[\mathsf{PrivK}_{A,\mathsf{OTP}} = 1] = \frac{1}{2}$$

3) By the simulation of $\mathsf{PrivK}_{A,\Pi}(n)$ by $D(G(x))$:

$$\mathbf{Pr}_{x \leftarrow U_n}[D(G(x)) = 1] = \mathbf{Pr}[\mathsf{PrivK}_{A,\Pi}(n) = 1]$$

Therefore

$$\mathbf{Pr}[\mathsf{PrivK}_{A,\Pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$

$\Longrightarrow$ $\mathbf{\Pi}$ (i.e. POTP) is EAV-secure. $\square$

Summary

- Proof that the pseudo OTP is secure...
- We have a provably secure scheme, rather than just a heuristic construction!

Summary

- Proof that the pseudo OTP is secure...
- ...with some caveats
  - Assuming $\boldsymbol{G}$ is a pseudorandom generator
  - Relative to our definition
- The only ways the scheme can be broken are:
  - If a weakness is found in $\boldsymbol{G}$
  - **If the definition isn't sufficiently strong** (next lecture!)

Have we gained anything?

- Yes! The **POTP has a key shorter than the message**
  - $\boldsymbol{n}$ bits vs. $\boldsymbol{p(n)}$ bits
- $\implies$ **Solved one of the limitations of the OTP**

- The fact that the parties internally generate a $\boldsymbol{p(n)}$-bit temporary string to encrypt/decrypt is irrelevant
- The key is what the parties share in advance
- Parties do not store the $\boldsymbol{p(n)}$-bit temporary value
- What about the other limitation? (next lectures)

End

Reference: Section 3.3.2