EE309 Advanced Programming Techniques for EE

Lecture 18: Pseudorandomness

INSU YUN (윤인수)

School of Electrical Engineering, KAIST

[Slides from Introduction to Cryptography -- MATH/CMSC 456 at UMD]

Core principles of modern crypto

- Formal definitions
  - Precise, mathematical model and definition of what security means
- Assumptions
  - Clearly stated and unambiguous
- Proofs of security
  - Move away from design-break-patch

Defining secure encryption

Crypto definitions (generally)

- Security guarantee/goal
  - What we want to achieve (or what we want to prevent the attacker from achieving)

- Threat model
  - What (real-world) capabilities the attacker is assumed to have

Recall

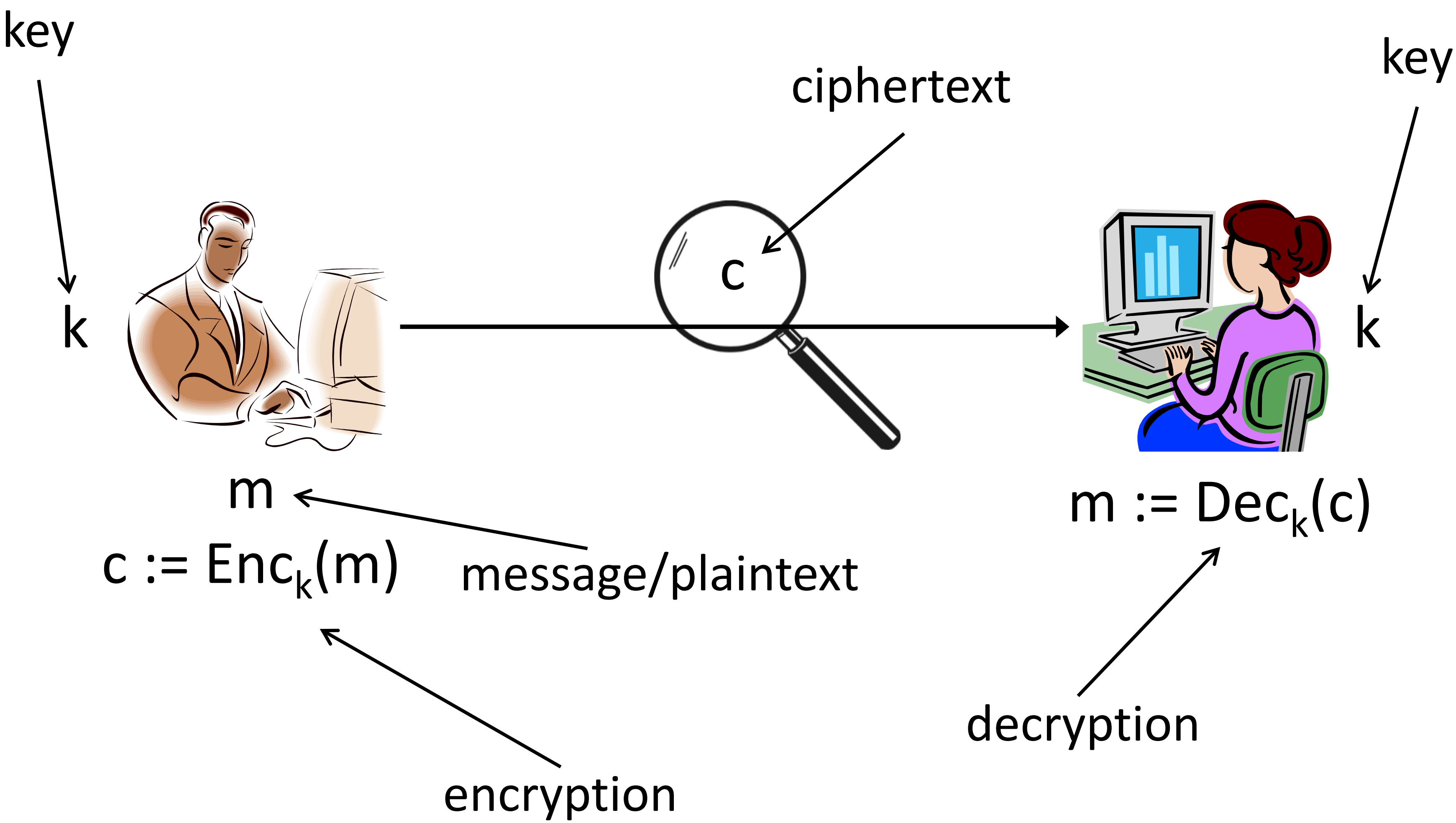
- A *private-key encryption scheme* is defined by a message space $\mathcal{M}$ and algorithms (Gen, Enc, Dec):
  - Gen (key-generation algorithm): generates k
  - Enc (encryption algorithm): takes key k and message $m \in \mathcal{M}$ as input; outputs ciphertext c

$$c \leftarrow \mathrm{Enc}_k(m)$$

  - Dec (decryption algorithm): takes key k and ciphertext c as input; outputs m.

$$m := \mathrm{Dec}_k(c)$$

Private-key encryption



Goal of secure encryption?

- How would you define what it means for encryption scheme (Gen, Enc, Dec) over message space $\mathcal{M}$ to be secure?
  - Against a (single) ciphertext-only attack

Secure encryption?

- “Impossible for the attacker to learn the key”
  - The key is a *means to an end*, not the end itself
  - Necessary (to some extent) but not sufficient
  - Easy to design an encryption scheme that hides the key completely, but is insecure
  - Can design schemes where most of the key is leaked, but the scheme is still secure

Secure encryption?

- “Impossible for the attacker to learn the plaintext from the ciphertext”
  - What if the attacker learns 90% of the plaintext?

The right definition

- "Regardless of any *prior* information the attacker has about the plaintext, the ciphertext should leak no *additional* information about the plaintext"
  - How to formalize?

Perfect secrecy

Probability review

- *Random variable (r.v.):* variable that takes on (discrete) values with certain probabilities

- Probability distribution for a r.v. specifies the probabilities with which the variable takes on each possible value
  - Each probability must be between 0 and 1
  - The probabilities must sum to 1

Probability review

- *Event*: a particular occurrence in some experiment
  - Pr[E]: probability of event E

- Conditional probability: probability that one event occurs, *given that* some other event occurred
  - Pr[A | B] = Pr[A and B]/Pr[B]

- Two random variables X, Y are *independent* if for all x, y: Pr[X=x | Y=y] = Pr[X=x]

Probability review

- Law of total probability: say $E_1, \ldots, E_n$ are a *partition* of all possibilities. Then for any A:

  $$\Pr[A] = \Sigma_i \Pr[A \text{ and } E_i] = \Sigma_i \Pr[A \mid E_i] \cdot \Pr[E_i]$$

- Bayes's theorem

  $$\Pr[A \mid B] = \Pr[B \mid A] \cdot \Pr[A]/\Pr[B]$$

Probability distributions

- Let M be the random variable denoting the value of the message
  - M ranges over $\mathcal{M}$
  - Context dependent!
  - Reflects the likelihood of different messages being sent, given the attacker's prior knowledge
  - E.g.,

$$\Pr[M = \text{“attack today”}] = 0.7$$
$$\Pr[M = \text{“don't attack”}] = 0.3$$

Probability distributions

- Fix some encryption scheme (Gen, Enc, Dec), and some distribution for M
- Consider the following (randomized) experiment:
  1. Generate a key k using Gen
  2. Choose a message m, according to the given distribution
  3. Compute $c \leftarrow \text{Enc}_k(m)$
- Let C be a random variable denoting the value of the ciphertext in this experiment
- This defines a distribution on the ciphertext!

Perfect secrecy (informal)

- "Regardless of any *prior* information the attacker has about the plaintext, the ciphertext should leak no *additional* information about the plaintext"

Perfect secrecy (formal)

- Encryption scheme (Gen, Enc, Dec) with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ is *perfectly secret* if for every distribution over $\mathcal{M}$, every m $\in \mathcal{M}$, and every c $\in \mathcal{C}$ with Pr[C=c] > 0, it holds that

$$\Pr[M = m \mid C = c] = \Pr[M = m].$$

- I.e., the distribution of M does not change conditioned on observing the ciphertext

Example 3

- Consider the shift cipher, and the distribution $\Pr[M = \text{'one'}] = ½$, $\Pr[M = \text{'ten'}] = ½$
- Take m = 'ten' and c = 'rqh'

- $\Pr[M = \text{'ten'} \mid C = \text{'rqh'}] = ?$

  $= 0$

  $\neq \Pr[M = \text{'ten'}]$

Conclusion

- The shift cipher is not perfectly secret!
  - At least not for 2-character messages

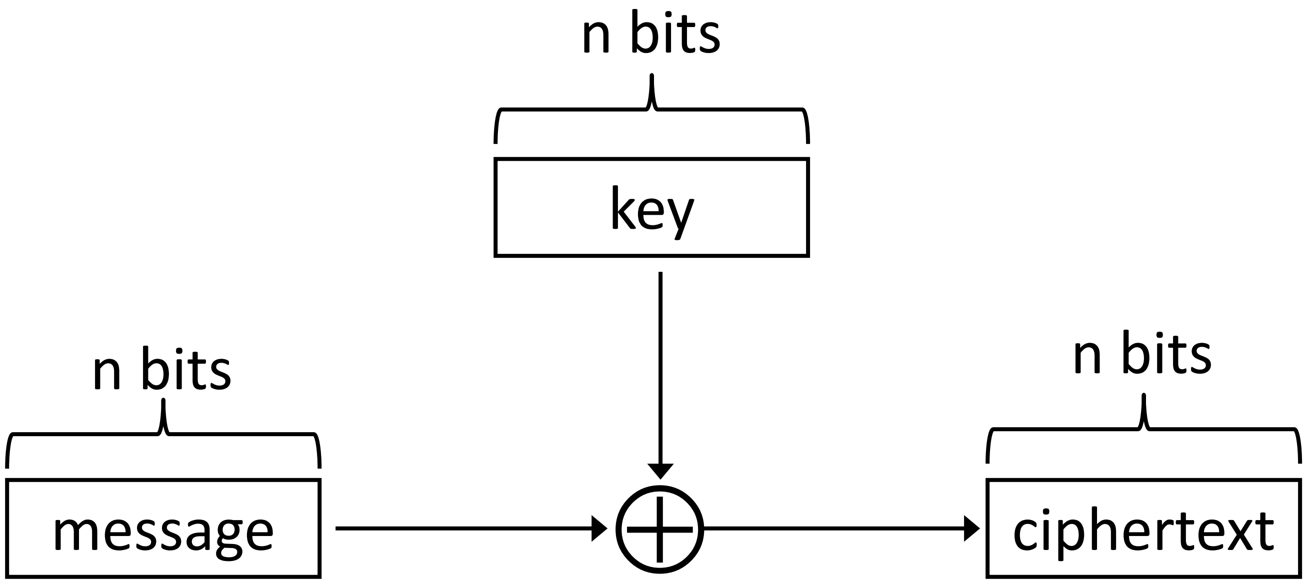
- How to construct a perfectly secret scheme?
  - One-time pad (proven by Shannon in 1949)

One-time pad

- Let $\mathcal{M} = \{0,1\}^n$
- Gen: choose a uniform key $k \in \{0,1\}^n$
- $\text{Enc}_k(m) = k \oplus m$
- $\text{Dec}_k(c) = k \oplus c$

- Correctness:
  $\text{Dec}_k(\ \text{Enc}_k(m)\ ) = k \oplus (k \oplus m)$
  $= (k \oplus k) \oplus m = m$

One-time pad



Perfect secrecy of one-time pad

- Fix arbitrary distribution over $\mathcal{M} = \{0,1\}^n$, and arbitrary $m, c \in \{0,1\}^n$
- $\Pr[M = m \mid C = c] = ?$

  $= \Pr[C = c \mid M = m] \cdot \Pr[M = m]/\Pr[C = c]$

- $\Pr[C = c]$

  $= \Sigma_{m'} \Pr[C = c \mid M = m'] \cdot \Pr[M = m']$

  $= \Sigma_{m'} \Pr[K = m' \oplus c \mid M = m'] \cdot \Pr[M = m']$

  $= \Sigma_{m'} 2^{-n} \cdot \Pr[M = m']$

  $= 2^{-n}$

Perfect secrecy of one-time pad

- Fix arbitrary distribution over $\mathcal{M} = \{0,1\}^n$, and arbitrary $m, c \in \{0,1\}^n$
- $\Pr[M = m \mid C = c] = ?$

$= \Pr[C = c \mid M = m] \cdot \Pr[M = m]/\Pr[C = c]$

$= \Pr[K = m \oplus c \mid M = m] \cdot \Pr[M = m] / 2^{-n}$

$= 2^{-n} \cdot \Pr[M = m] / 2^{-n}$

$= \Pr[M = m]$

One-time pad

- The one-time pad achieves perfect secrecy!
- One-time pad has historically been used in the real world
  - E.g., “red phone” between DC and Moscow
- Not currently used!
  - Why not?

One-time pad

- Several limitations
  - The key is as long as the message
  - Only secure if each key is used to encrypt a *single* message
    - (Trivially broken by a known-plaintext attack)

$\Rightarrow$ Parties must share keys of (total) length equal to the (total) length of all the messages they might ever send

Optimality of the one-time pad

- Theorem: if (Gen, Enc, Dec) with message space $\mathcal{M}$ is perfectly secret, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Where do we stand?

- We defined the notion of perfect secrecy
- We proved that the one-time pad achieves it!
- We proved that the one-time pad is optimal!
  - I.e., we cannot improve the key length
- Are we done?

- Do better *by relaxing the definition*
  - But in a meaningful way…

Perfect secrecy

- Requires that *absolutely no information* about the plaintext is leaked, even to eavesdroppers *with unlimited computational power*
  - Has some inherent drawbacks
  - Seems unnecessarily strong

Computational secrecy

- Would be ok if a scheme leaked information *with tiny probability* to eavesdroppers *with bounded computational resources*
- I.e., we can relax perfect secrecy by
  - Allowing security to "fail" with tiny probability
  - Restricting attention to "efficient" attackers

Bounded attackers?

- Consider brute-force search of key space; assume one key can be tested per clock cycle
- Desktop computer $\approx 2^{57}$ keys/year
- Supercomputer $\approx 2^{80}$ keys/year
- Supercomputer since Big Bang $\approx 2^{112}$ keys
  - Restricting attention to attackers who can try 2^{112} keys is fine!

- Modern key space: 2^{128} keys or more…

Roadmap

- We will give an alternate (but equivalent) definition of perfect secrecy
  - Using a randomized experiment
- That definition has a natural relaxation

- **Warning**: the material gets much more difficult now

Perfect indistinguishability

- Let Π=(Gen, Enc, Dec) be an encryption scheme with message space $\mathcal{M}$, and A an adversary
- Define a randomized exp't $\mathrm{PrivK}_{A,\Pi}$:
  1. A outputs $m_0, m_1 \in \mathcal{M}$
  2. $k \leftarrow$ Gen, $b \leftarrow \{0,1\}$, $c \leftarrow \mathrm{Enc}_k(m_b)$
  3. $b' \leftarrow A(c)$

  Challenge ciphertext

  Adversary A *succeeds* if $b = b'$, and we say the experiment evaluates to 1 in this case

Perfect indistinguishability

- Easy to succeed with probability ½ …
- Π is *perfectly indistinguishable* if for all attackers (algorithms) A, it holds that

$$\Pr[\mathrm{PrivK}_{A,\Pi} = 1] = 1/2$$

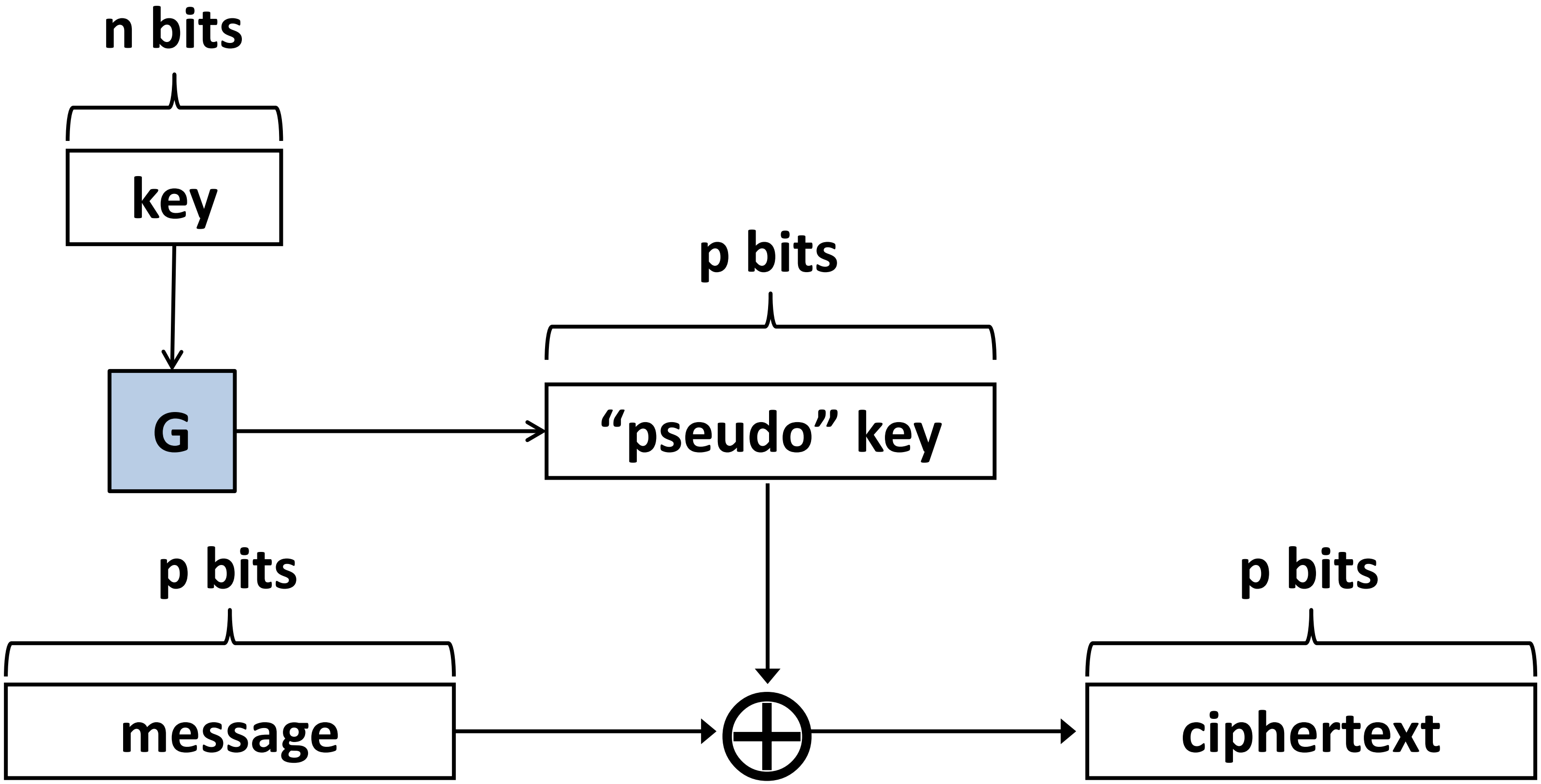
Computational indistinguishability (concrete version)

- Π is (t, ε)-*indistinguishable* if for all attackers A running in time at most t, it holds that

$$\Pr[\mathrm{PrivK}_{A,\Pi} = 1] \leq ½ + \varepsilon$$

- Note: $(\infty, 0)$-indistinguishable = perfect indistinguishability
  - Relax definition by taking $t < \infty$ and $\varepsilon > 0$

“Pseudo” one-time pad (i.e., Stream cipher)



Pseudorandomness

Pseudorandomness

- Important building block for computationally secure encryption

- Important concept in cryptography

What does “random” mean?

- What does “uniform” mean?
- Which of the following is a uniform string?
  - 0101010101010101
  - 0010111011100110
  - 0000000000000000
- If we generate a uniform 16-bit string, each of the above occurs with probability 2^{-16}

What does "uniform" mean?

- "Uniformity" is not a property of a *string*, but a property of a *distribution*

- A distribution on *n*-bit strings is a function $D: \{0,1\}^n \to [0,1]$ such that $\Sigma_x D(x) = 1$
  - The *uniform* distribution on *n*-bit strings, denoted U_n, assigns probability 2^{-n} to every $x \in \{0,1\}^n$

What does “pseudorandom” mean?

- Informal: cannot be distinguished from uniform (i.e., random)
- Which of the following is pseudorandom?
  - 0101010101010101
  - 0010111011100110
  - 0000000000000000
- Pseudorandomness is a property of a *distribution*, not a *string*

Pseudorandomness (take 1)

- Fix some distribution D on *n*-bit strings
  - $x \leftarrow D$ means "sample x according to D"
- Historically, D was considered pseudorandom if it "passed a bunch of statistical tests"
  - $\Pr_{x \leftarrow D}[1^{st}$ bit of x is $1] \approx 1/2$
  - $\Pr_{x \leftarrow D}[$parity of x is $1] \approx 1/2$
  - $\Pr_{x \leftarrow D}[\mathrm{Test}_i(x)=1] \approx \Pr_{x \leftarrow U_n}[\mathrm{Test}_i(x)=1]$ for $i = 1, \ldots$

Pseudorandomness (take 2)

- This is not sufficient in an adversarial setting!
  - Who knows what statistical test an attacker will use?

- Cryptographic def'n of pseudorandomness:
  - D is pseudorandom if it passes <u>all</u> *efficient* statistical tests

Pseudorandomness (concrete)

- Let D be a distribution on *p*-bit strings

- D is (t, ε)-pseudorandom if for all A running in time at most t,

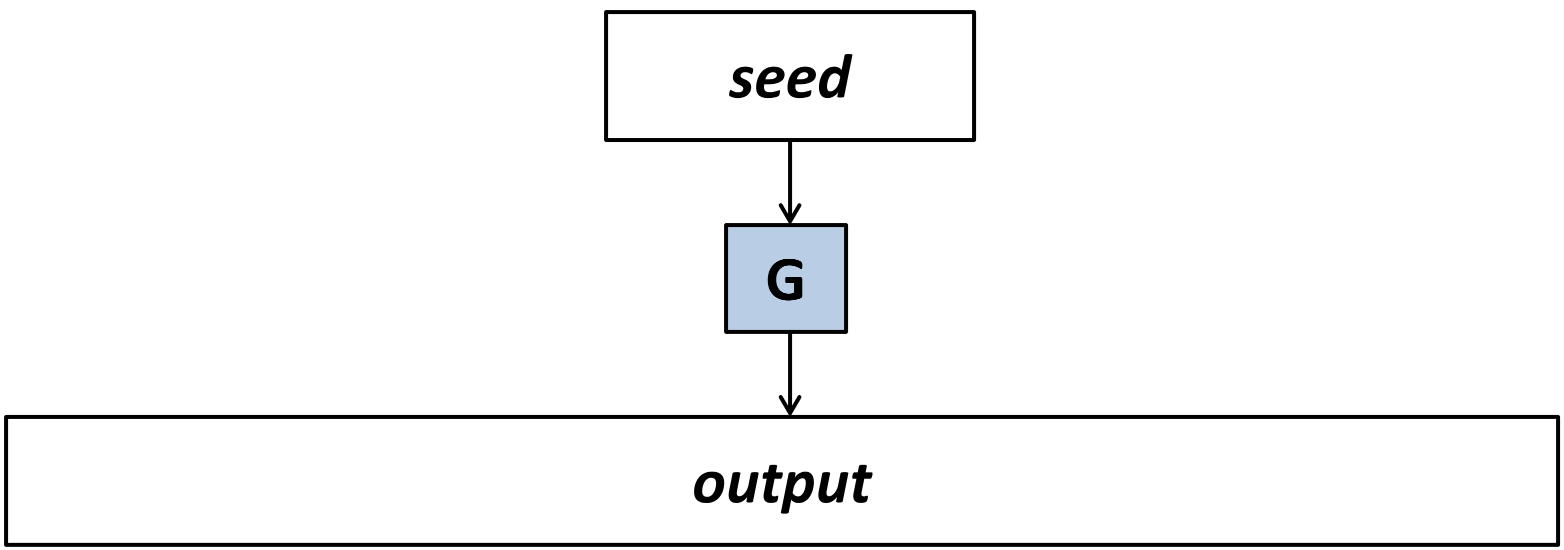
$$| \Pr_{x \leftarrow D}[A(x)=1] - \Pr_{x \leftarrow U_p}[A(x)=1] | \leq \varepsilon$$

Pseudorandom generators (PRGs)

- A PRG is an efficient, deterministic algorithm that expands a *short, uniform seed* into a *longer, pseudorandom* output
  - Useful whenever you have a “small” number of true random bits, and want lots of “random-looking” bits

PRGs

- Let G be a deterministic, poly-time algorithm that is *expanding*, i.e., $|G(x)| = p(|x|) > |x|$



PRGs

- G is a PRG iff $\{D_n\}$ is pseudorandom
  - D_n = the distribution on p(n)-bit strings defined by choosing $x \leftarrow U_n$ and outputting G(x)

- I.e., for all efficient distinguishers A, there is a negligible function ε such that

  $$| \Pr_{x \leftarrow U_n}[A(G(x))=1] - \Pr_{y \leftarrow U_{p(n)}}[A(y)=1] | \leq \varepsilon(n)$$

- I.e., no efficient A can distinguish whether it is given G(x) (for uniform x) or a uniform string y!

Example (insecure PRG)

- Let G(x) = 0….0
  - Distinguisher?
  - Analysis?

Example (insecure PRG)

- Let G(x) = x | OR(bits of x)
  - Distinguisher?
  - Analysis?

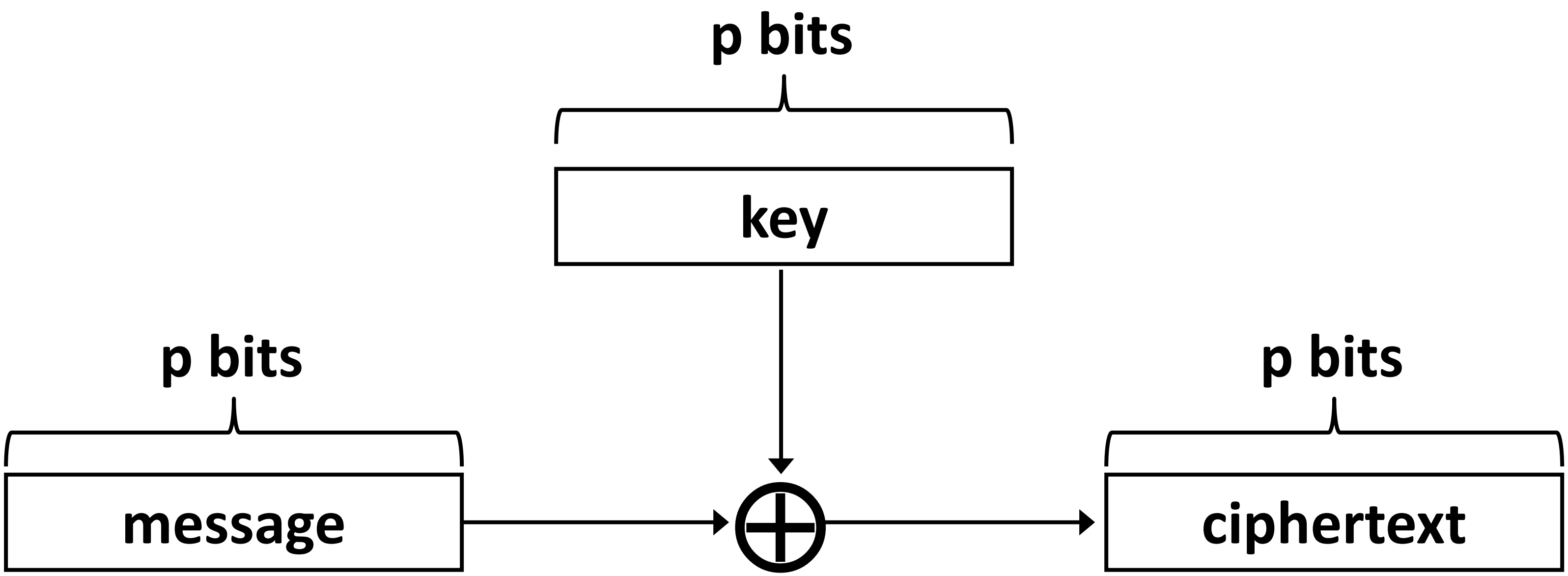
Do PRGs exist?

- We don't know…
  - Would imply P $\neq$ NP
- We will *assume* certain algorithms are PRGs
  - Recall the 3 principles of modern crypto…
  - This is what is done in practice

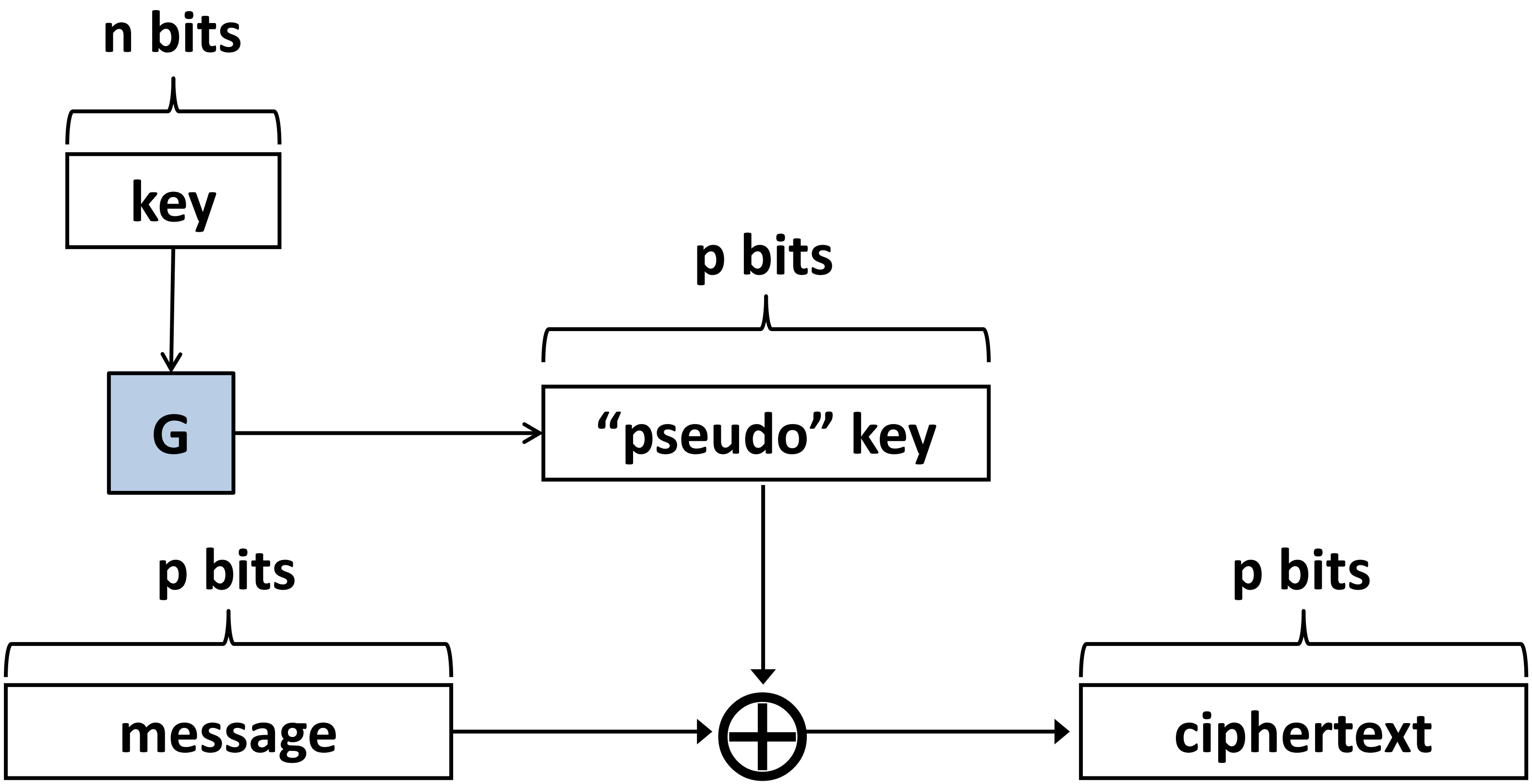
Where things stand

- We saw that there are some inherent limitations if we want perfect secrecy
  - In particular, key must be as long as the message
- We defined computational secrecy, a relaxed notion of security
- Can we overcome prior limitations?

Recall: one-time pad



"Pseudo" one-time pad



Pseudo one-time pad

- Let G be a deterministic algorithm, with $|G(k)| = p(|k|)$
- $Gen(1^n)$: output uniform n-bit key k
  - Security parameter n $\Rightarrow$ message space $\{0,1\}^{p(n)}$
- $Enc_k(m)$: output $G(k) \oplus m$
- $Dec_k(c)$: output $G(k) \oplus c$

- Correctness is obvious…

Security of pseudo-OTP?

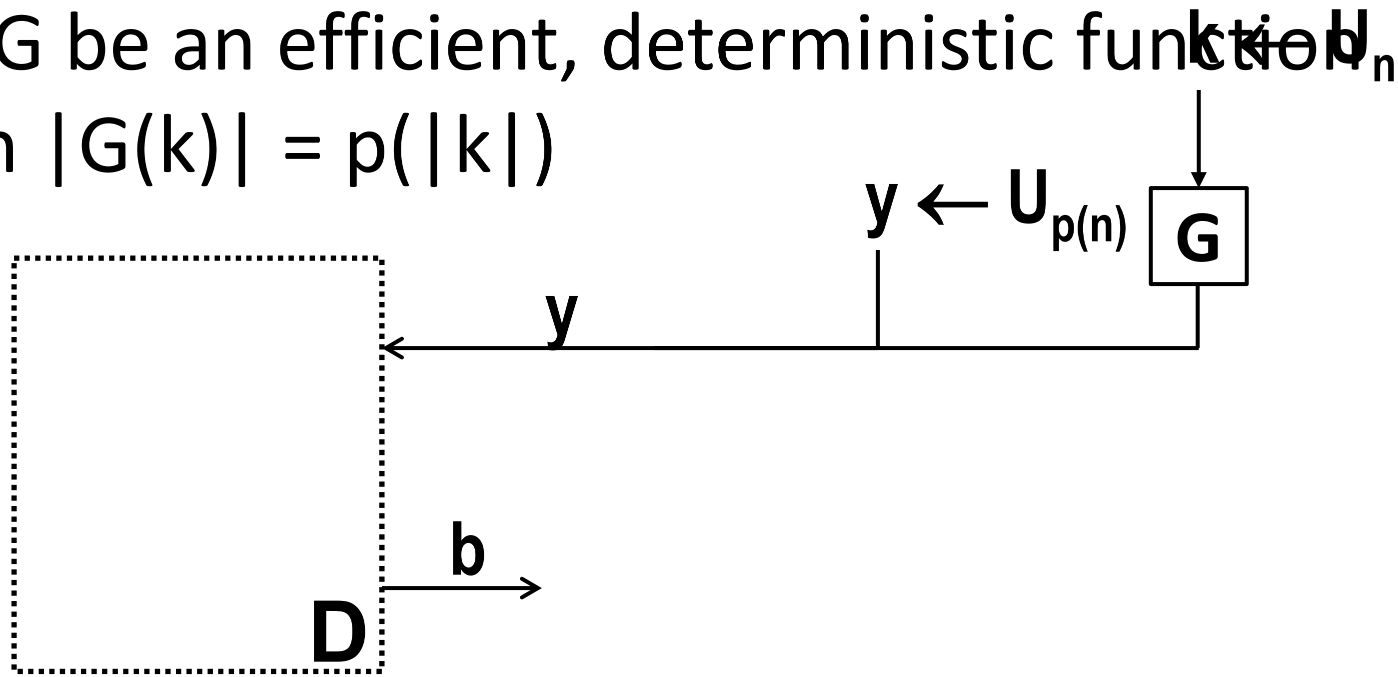
- Would like to be able to *prove* security
  - Based on the *assumption* that G is a PRG

Definitions, proofs, and 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 G
- *Can* prove security based on *the assumption* that G is a pseudorandom generator

PRGs, revisited

- Let G be an efficient, deterministic function with $|G(k)| = p(|k|)$



For any efficient D, the probabilities that D outputs 1 in each case must be “close”

Proof by reduction

1. Assume G is a pseudorandom generator
2. Assume toward a contradiction that there is an efficient attacker A who "breaks" the pseudo-OTP scheme (as per the definition)
3. Use A as a subroutine to build an efficient D that "breaks" pseudorandomness of G
   – By assumption, no such D exists!

   $\Rightarrow$ No such A can exist

Alternately…

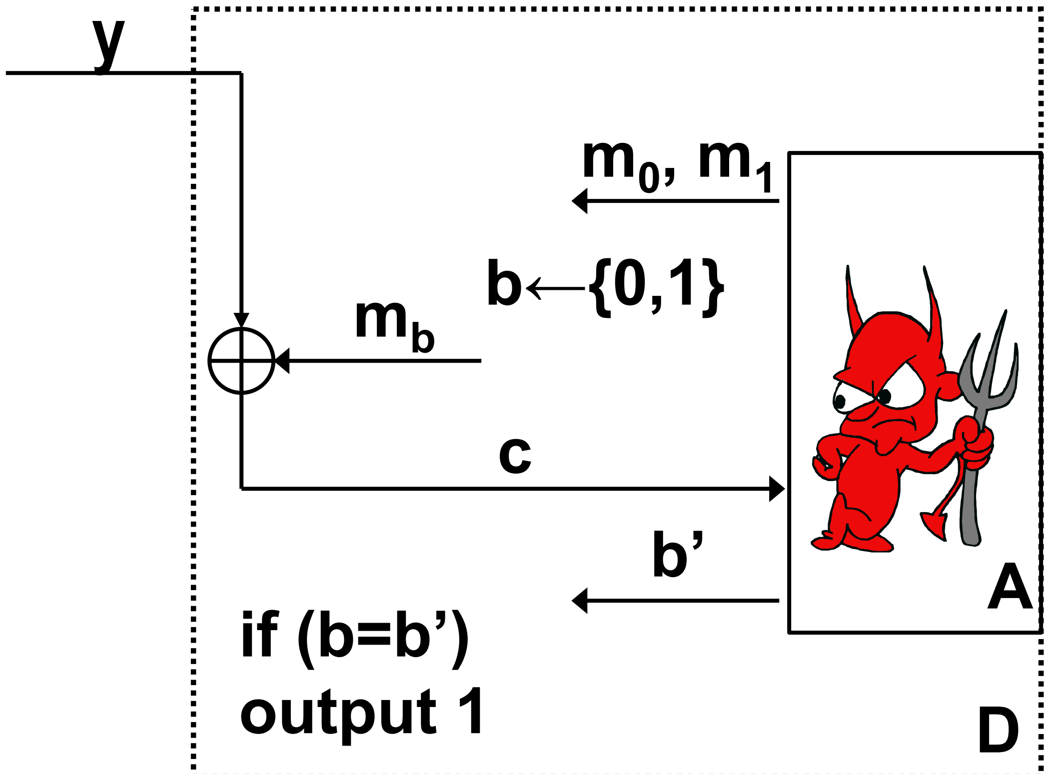
1. Assume G is a pseudorandom generator
2. Fix some arbitrary, efficient A attacking the pseudo-OTP scheme
3. Use A as a subroutine to build an efficient D attacking G
   – Relate the distinguishing gap of D to the success probability of A
4. By assumption, the distinguishing gap of D must be negligible

$\Rightarrow$ Use this to bound the success probability of A

Security theorem

- If G is a pseudorandom generator, then the pseudo one-time pad Π is EAV-secure (i.e., computationally indistinguishable)

The reduction



Analysis

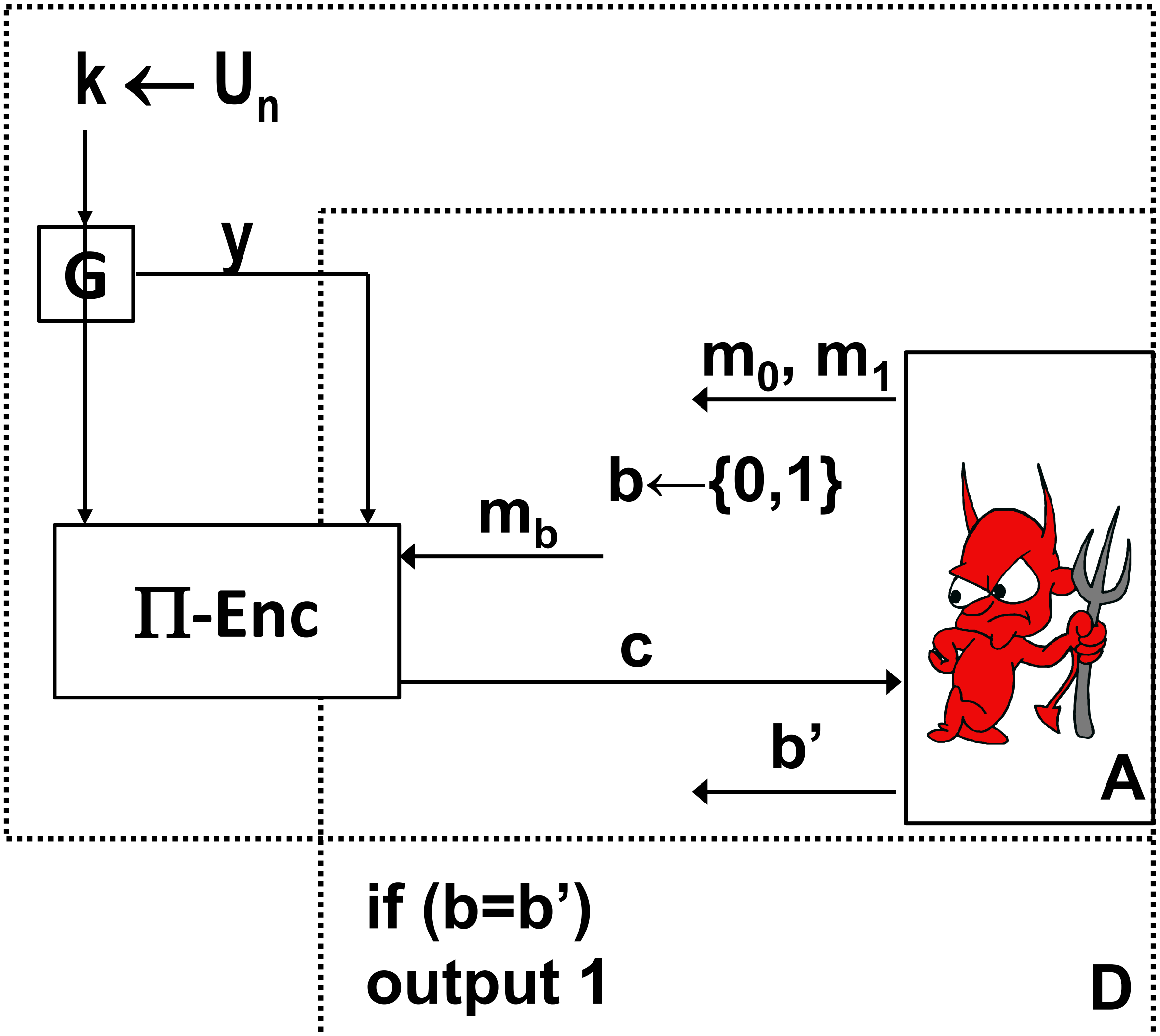
- If A runs in polynomial time, then so does D

Analysis

- Let $\mu(n) = \Pr[\mathrm{PrivK}_{A,\Pi}(n) = 1]$
- Claim: when $y=G(x)$ for uniform x, then the view of A is *exactly* as in $\mathrm{PrivK}_{A,\Pi}(n)$

  $\Rightarrow \Pr_{x \leftarrow U_n}[D(G(x))=1] = \mu(n)$

The reduction



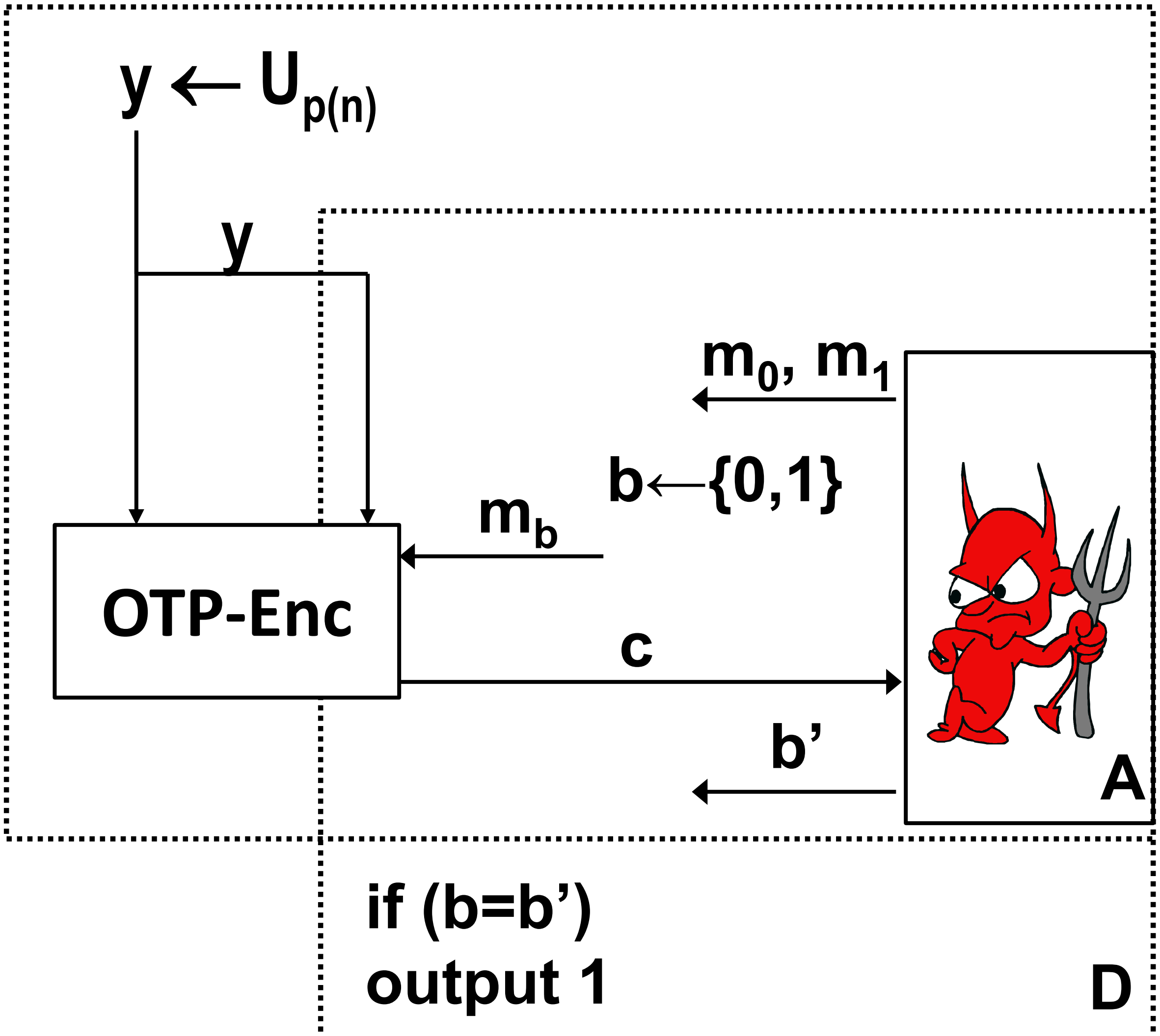
Analysis

- Let $\mu(n) = \Pr[\mathrm{PrivK}_{A,\Pi}(n) = 1]$
- If $y=G(x)$ for uniform x, then the view of A is *exactly* as in $\mathrm{PrivK}_{A,\Pi}(n)$

  $\Longrightarrow \Pr_{x \leftarrow U_n}[D(G(x))=1] = \mu(n)$
- If distribution of y is uniform, then A succeeds with probability exactly ½

  $\Longrightarrow \Pr_{y \leftarrow U_{p(n)}}[D(y)=1] = ½$

The reduction



Analysis

- Let $\mu(n) = \Pr[\mathrm{PrivK}_{A,\Pi}(n) = 1]$
- If $y=G(x)$ for uniform x, then the view of A is *exactly* as in $\mathrm{PrivK}_{A,\Pi}(n)$

  $\Rightarrow \Pr_{x \leftarrow U_n}[D(G(x))=1] = \mu(n)$
- If distribution of y is uniform, then A succeeds with probability exactly ½

  $\Rightarrow \Pr_{y \leftarrow U_{p(n)}}[D(y)=1] = ½$
- Since G is pseudorandom:

$$| \mu(n) - ½ | \leq \mathrm{negl}(n)$$

  $\Rightarrow \Pr[\mathrm{PrivK}_{A,\Pi}(n) = 1] \leq ½ + \mathrm{negl}(n)$

Have we gained anything?

- YES: the pseudo-OTP has a key shorter than the message
  – n bits vs. p(n) bits
- The fact that the parties *internally* generate a p(n)-bit temporary string to encrypt/decrypt is **irrelevant**
  – The *key* is what the parties share *in advance*
  – Parties do not store the p(n)-bit temporary value

Recall…

- Perfect secrecy has two limitations/drawbacks
  - Key as long as the message
  - Key can only be used once

- We have seen how to circumvent the first
- Does the pseudo OTP have the second limitation?

- How can we circumvent the second?

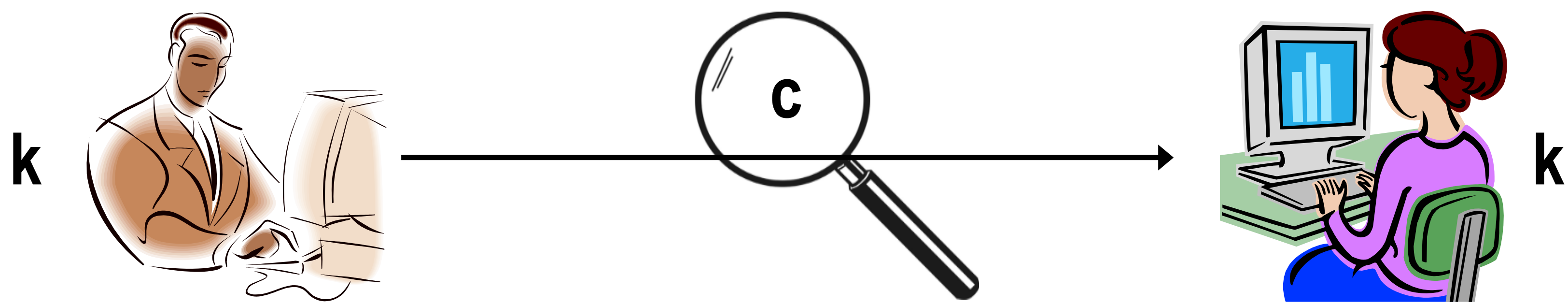
But first…

- Develop an appropriate security definition

- Recall that security definitions have two parts
  - Security goal
  - Threat model

- We will keep the security goal the same, but strengthen the threat model

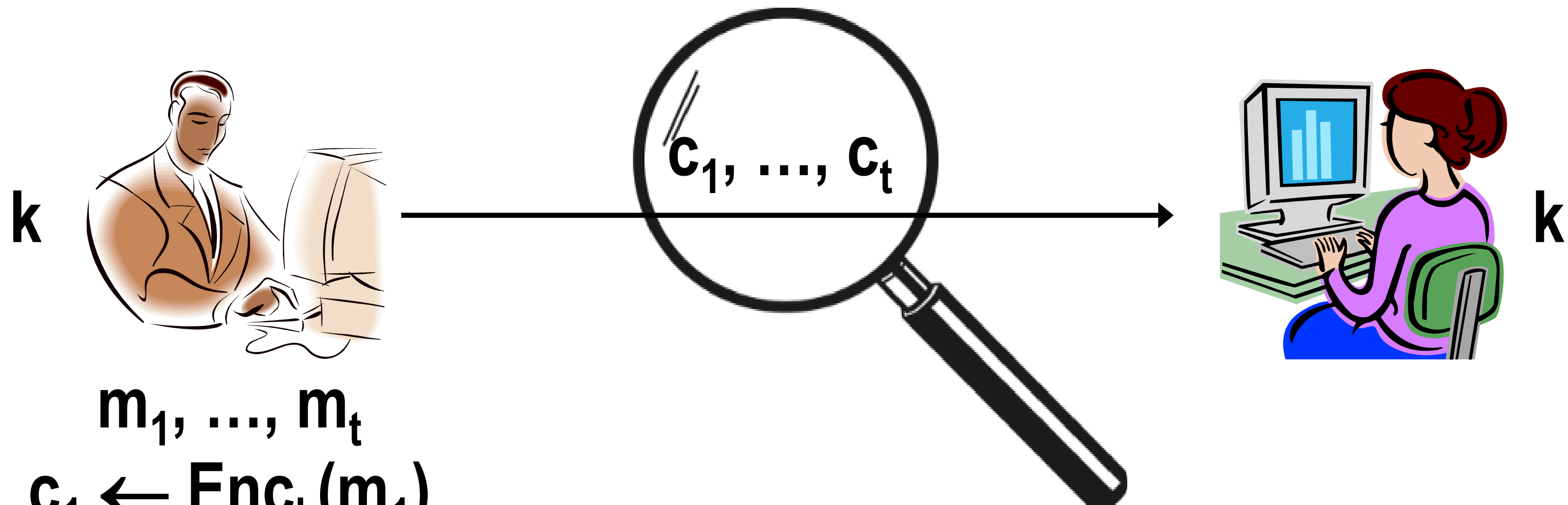
Single-message secrecy



m

$c \leftarrow \mathrm{Enc}_k(m)$

Multiple-message secrecy



$m_1, \ldots, m_t$

$c_1 \leftarrow \mathrm{Enc}_k(m_1)$

…

$c_t \leftarrow \mathrm{Enc}_k(m_t)$

A formal definition

- Fix Π, A
- Define a randomized exp't $\text{PrivK}^{\text{mult}}_{A,\Pi}(n)$:
  1. $A(1^n)$ outputs two **vectors** $(m_{0,1}, \ldots, m_{0,t})$ and $(m_{1,1}, \ldots, m_{1,t})$
     - Require that $|m_{0,i}| = |m_{1,i}|$ for all i
  2. $k \leftarrow \text{Gen}(1^n)$, $b \leftarrow \{0,1\}$, for all i: $c_i \leftarrow \text{Enc}_k(m_{b,i})$
  3. $b' \leftarrow A(c_1, \ldots, c_t)$; A *succeeds* if $b = b'$, and experiment evaluates to 1 in this case

A formal definition

- Π is *multiple-message indistinguishable* if for all PPT attackers A, there is a negligible function ε such that

$$\Pr[\mathrm{PrivK}^{\mathrm{mult}}_{A,\Pi}(n) = 1] \leq \tfrac{1}{2} + \varepsilon(n)$$

- Exercise: show that the pseudo-OTP is *not* multiple-message indistinguishable