Cryptography – Computational Secrecy

Day 5

Try Question 1

Python Lab 2

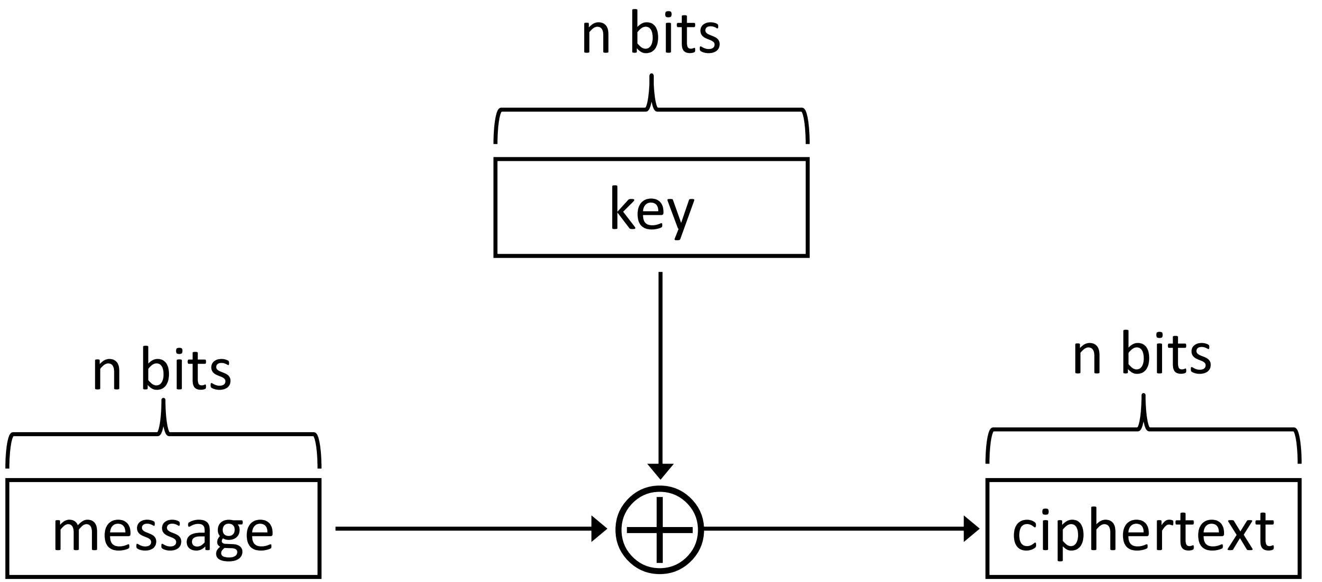
Review

One-time pad

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

- Correctness:
  $Dec_k(Enc_k(m)) = m$

One-time pad



One-time pad

- We defined the notion of perfect secrecy
- The one-time pad achieves perfect secrecy!
- One-time pad is Optimal!
  - **Thm.** If (Gen, Enc, Dec) with message space $\mathcal{M}$ is perfectly secret, then $|\mathcal{K}| \geq |\mathcal{M}|$.
  - I.e., we cannot improve the key length

One-time pad

- Drawbacks of One-time Pad
  - Key as long the message
  - Only secure if each key is used to encrypt *once*
  - Trivially broken by a known-plaintext attack

- These limitations are *inherent* for schemes achieving perfect secrecy

Perfect Secrecy

- Drawbacks of Perfect Secrecy
  - Key as long the message
  - Only secure if each key is used to encrypt *once*

- Are we done?

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

Computational secrecy

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

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

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

Tiny probability of failure?

- Say security fails with probability 2^{-60}
  - Should we be concerned about this?
  - With probability > 2^{-60}, the sender and receiver will both be struck by lightning in the next year…
  - Something that occurs with probability 2^{-60}/sec is expected to occur once every 100 billion years

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

Perfect indistinguishability

- Π = (Gen, Enc, Dec), message space $\mathcal{M}$
- Informally:
  - Two messages m_0, m_1; one is chosen and encrypted (using unknown k) to give $c \leftarrow \text{Enc}_k(m_b)$
  - Adversary A is given c and tries to determine which message was encrypted
  - Π is perfectly indistinguishable if *no* A can guess correctly with probability *any better than ½*

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 $\text{PrivK}_{A,\Pi}$:
  1. A outputs $m_0, m_1 \in \mathcal{M}$
  2. $k \leftarrow$ Gen, $b \leftarrow \{0,1\}$, $c \leftarrow \text{Enc}_k(m_b)$
  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 …

- Scheme Π is *perfectly indistinguishable* if for all attackers (algorithms) A, it holds that

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

Perfect indistinguishability

- Claim: Π is perfectly indistinguishable $\Leftrightarrow$ Π is perfectly secret

- I.e., perfect indistinguishability is just an alternate definition of perfect secrecy

Try Question 2

Computational secrecy?

- Idea: relax perfect indistinguishability

- Two approaches
  - Concrete security
  - Asymptotic security

Computational indistinguishability (concrete version)

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

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

  - Relax definition by taking $t < \infty$ and $\varepsilon > 0$

Concrete security

- Parameters t, ε are what we ultimately care about in the real world
- Does not lead to a clean theory...
  - Sensitive to exact computational model
  - Π can be (t, ε)-secure for many choices of t, ε
- Would like to have schemes where users can adjust the achieved security as desired

Asymptotic security

- Introduce *security parameter* n
  - For now, think of n as the key length
  - Chosen by honest parties when they generate/share key
    - Allows users to tailor the security level
  - Known by adversary

- Measure running times of all parties, and the success probability of the adversary, as functions of n

Computational indistinguishability (asymptotic)

- Computational indistinguishability:
  - Security may fail with probability *negligible in n*
  - Restrict attention to attackers running in time (at most) *polynomial in n*

Try Question 3