COMPGA4 Crypto II, 2008

Lectures 1–5

Helger Lipmaa

University College London

COMPGA04 Crypto II, Lectures 1–5

Outline I

1. Lecture 1
   - Intro to Intro
   - Secret-key cryptosystems
   - Game-Based Security
   - IND-KPA, IND-CPA, IND-CCA Security
   - Reduction
   - f-OTP
2. Lecture 2
   - IND-CPA of PKC
   - Preliminaries: Algebra
   - DL Assumption
   - Elgamal Cryptosystem
   - CDH And DDH Assumptions

Outline II

   - IND-CPA Security of Elgamal
3. Lecture 3
   - IND-CPA Secret-Key Cryptosystems
   - Counter Mode
   - PRPs/PRFs
   - Proof: CTR Is Secure
   - Cipher-Block Chaining Mode
4. Lecture 4
   - PRP to PRF, PRF to PRP
5. Lecture 5
   - Authentication
   - Message Authentication Codes
   - Wegman-Carter MAC

Lecture 1

Goals of The Course

- Introduction to modern cryptography
- For people who are oriented towards
  - Academic career
  - Industrial career: position where one must understand what does it mean to be secure, is able to choose secure primitives/protocols, verify their security and possible design new secure ones
- Emphasis on proofs: how to prove that something is secure
- Emphasis on provably secure primitives/protocols: not on the ones in standards
  - More precisely, provably secure and efficient
- After course, you should be able to follow modern academic literature on subject

What Is Modern Cryptography?

- (Old) cryptography: building secure codes for encryption
- (Old) Security objective: confidentiality (not well-defined)
- (Old) Security: a code is secure until nobody has broken it
- Cryptography (modern): building provably secure protocols for securing arbitrary operations on Internet
- Security objective (modern): many different, precisely defined
  - confidentiality, authenticity, integrity, robustness, ...
- Security (modern): provable security
- Modern cryptography: science of designing provably secure cryptographic protocols

Security: Classical World

- For more than 2000 years, mostly governments used cryptography for secure information exchange with their spies, diplomats, ..., but also with other governments
- Lifes and colossal wealth depended upon security of used codes
- ...yet most of codes were broken sometimes centuries before their use ended

Security: Classical World

- Codes themselves were kept secret
  - Security by obscurity: it is arguably more difficult to break code you don't know
  - If your code actually is more secure, you don't want your adversaries to start using it
- Attacks were kept secret
  - You want your adversary to keep using code you have already broken
  - You don't want your adversary to break your own codes by using your own attack methods

Birth of Modern Cryptography: Kerckhoffs

Auguste Kerckhoffs [1883] made some important comments that are all relevant even now:

- The system should be, if not theoretically unbreakable, unbreakable in practice
    - “Computational security” — (practically) all contemporary systems are computationally secure
- The design of a system should not require secrecy and compromise of the system should not inconvenience the correspondents (Kerckhoffs’ principle)
    - Very important: a protocol should stay secure even if the adversary knows it

(To be continued)

Birth of Modern Cryptography: Kerckhoffs

(continued)

- The key should be memorable without notes and should be easily changeable
    - Current translation: keys should be short enough to fit into secure hardware
- The cryptograms should be transmittable by telegraph
    - Current translation: they should be (short) bitstrings ☺
- The apparatus or documents should be portable and operable by a single person
    - Current translation: it should be easy to implement and operate cryptographic software by a layman

Birth of Modern Cryptography: WW2

- WW2 and the concurrent tense political situation stimulated new research in cryptography
- Breaking the German/Japanese codes helped the allies to won the World War II a few years earlier
- Cryptographic research was top secret for the same reasons it had already been
    - Breaking of Enigma was made public only 40 years later!
    - Up to then, previous British colonies happily used war-time Enigmas, assuming that they are secure. . .
    - The first computer — Colossus — was specifically built to decrypt German codes. After the war it was dismantled, and even its existence was made top secret
- One important published paper: [Shannon, 1949]

Birth of Modern Cryptography: Shannon’s work

- Shannon studied the properties of ciphers, that are used to encrypt plaintexts by using some keys
- He made some fundamental observations:
    - **Theorem 1.** One-time pad is secure
    - **Theorem 2.** You cannot build anything more efficient than one-time pad that is (as) secure
- Security here is information-theoretical: nothing will leak even to omnipotent adversaries

Birth of Modern Cryptography: Shannon's work

- Recall that one-time pad needs one to use one-time keys, and the total bitlength of keys is the same as the total bitlength of encrypted messages
- To overcome Theorem 2, Shannon also studied computational security, proposing some basic ideas (confusion, diffusion, product ciphers) that are even used nowadays in the design of ciphers
- Shannon did not define computational security: this had to wait for the birth of the complexity theory in 60s/70s

Birth of Modern Cryptography: Sixties

- (Re)invention of computers in later 40s, their decrease in size and the increase of their availability also meant that one now needed cryptography to protect also commercial data, not only military data
- Because there's a lot of more commercial data — and most of it is secret but not top secret —, cryptographic research was ready to go public

Birth of Modern Cryptography: Seventies

- Public competition to design a standard cipher for nonclassified data — won by DES
- Discovery of public-key cryptography in the academic community: [Diffie and Hellman, 1976, Rivest et al., 1978]
  - PKC was independently and slightly before discovered in the secret agencies. This was published 20 years later, and many details are still missing
- However, none of the first schemes are secure according to the contemporary definitions

Why Is RSA Not Secure?

- Recall that in RSA, the ciphertext is computed as $y = x^e \pmod{n}$ for plaintext x, public key e
- RSA has the homomorphic property: $x_1^e \cdot x_2^e \equiv (x_1 x_2)^e \mod n$
- Thus, adversary can compute a ciphertext of $x_1 x_2$ by herself
- RSA is malleable: knowing a ciphertext of some plaintext, one can generate a ciphertext of a related plaintext
- **Example:** given a ciphertext of "Yes", compute a ciphertext of "Not", and vice versa
- **Example:** given opponent's secret bid x on auction, compute bid of $x + 1$
- Note: homomorphic properties are highly desired in some applications

Modern Cryptography: 1982+

- Modern cryptography is ripe discipline dealing with secure communications
  - Personal opinion: symmetric cryptography (block ciphers, stream ciphers, ...) is not ripe, and thus we will not deal with it in this course
- MC enforces strict discipline when you construct secure solution for cryptographic problem:
  - Study problem. Define security in this setting
  - Design a protocol
  - Prove that this protocol is secure according to definition
- Often, first secure protocol is inefficient but it demonstrates feasibility
- End goal is to design something that is both efficient, and secure with respect to strongest reasonable security definition

Secret-Key Cryptosystems: Syntactic Definition

Secret-key cryptosystem is a triple of three algorithms:

- Key generation $G(k)$ that outputs a k-bit random string
- Encryption $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$
- Decryption $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

We require that $D_{\mathsf{sk}}(E_{\mathsf{sk}}(m)) = m$ for every sk, m

Secret-Key Cryptosystems: Bad Examples

- **Variant 1.** A cryptosystem is secure if, given ciphertext, we cannot find key
  - But then $E_{\mathsf{sk}}(m) := m$ is a secure cipher!
- **Variant 2.** A cryptosystem is secure if, given ciphertext, we cannot find plaintext
  - But what if we are only interested in finding the first byte of the plaintext (“Yes” vs “Not”)?
- **Variant 3.** A cryptosystem is secure if, given ciphertext, we cannot find a single bit of plaintext
  - But if we know that the plaintext is either “Yes” or ”Not” then XOR of two bits may reveal which one was encrypted
- General idea of correct definition: adversary knows that ciphertext encrypts one of two messages. Let her guess which one was encrypted

Game-Based Security

- We define similar security notions by using games
- Game is interactive protocol between challenger and adversary
- Challenger has access to internals of cryptographic protocol:
  - He generates random keys
  - ...and performs all operations that require secret information
- Adversary:
  - Obtains public information from challenger
  - Has black-box access to primitive by making game-dependent queries to challenger
- At some point in game, adversary receives challenge
- She wins when at end, she answers correctly to the challenge
- We say the protocol is (τ, ε)-secure wrt this game if no τ-time adversary wins the game with probability $\geq \Pr[\text{winning without participating in game}] + \varepsilon$

Security Notion = Adversarial Goal + Adversarial Capabilities

- Protocol is secure if adversary does not "break" the game
- Game = adversarial goal
  - Defined by the challenge, and by what it means to answer correctly
- Adversarial capabilities:
  - Qualitative: what kind of access the adversary has to the system
  - Quantitative: time, number of queries, success probability

Secret-Key Cryptosystems: IND-KPA Security

IND-KPA (indistinguishable against known plaintext attacks) **game**:

1. **Setup phase:** Challenger generates random key $\mathsf{sk} \leftarrow G(k)$
2. **Challenge phase:**
   - Adversary choses two messages (m_0^*, m_1^*) of the same length, and sends them to challenger
   - Challenger choses a random bit $b \leftarrow \{0, 1\}$. She sends $c^* \leftarrow E_{\mathsf{sk}}(m_b^*;)$ to adversary
3. **Guessing phase:** Adversary outputs a bit b^*. She wins if $b^* = b$

We say that a secret-key cryptosystem is (τ, ε)-IND-KPA secure if $\Pr[\text{Adversary wins}] < \frac{1}{2} + \varepsilon$ for any adversary that works in time τ
Goal: IND, quantitative capability: (τ, ε), qualitative capability: KPA.

Recall: One-Time Pad

Recall from Crypto I. One-time pad is defined as follows:

- $G(k)$ returns a k-bit string.
- $E : \{0, 1\}^k \times \{0, 1\}^k \to \{0, 1\}^k$ returns $E_{\mathsf{sk}}(m) := \mathsf{sk} \oplus m$.
- $D : \{0, 1\}^k \times \{0, 1\}^k \to \{0, 1\}^k$ returns $D_{\mathsf{sk}}(c) := \mathsf{sk} \oplus c$.

Fact [Shannon, 1949]. OTP is secure.

One-Time Pad Is IND-KPA Secure

Theorem

One-time pad is $(\infty, 0)$-IND-KPA secure.

Note: Every adversary has success probability at least $\frac{1}{2}$. Thus this theorem claims all adversaries have success probability exactly $\frac{1}{2}$.

One-Time Pad Is IND-KPA Secure

Proof.

During game, challenger creates $sk \leftarrow \{0,1\}^k$. Adversary sends $m_0^*, m_1^* \in \{0,1\}^k$ to challenger, who replies with $m_b^* \oplus sk$ for random $b \leftarrow \{0,1\}$. Adversary has to guess b. For any bitstring c^* that adversary sees, and for both i,

$$\Pr_{sk}[m_i^* = c^* \oplus sk] = \Pr_{sk}[sk = \underbrace{m_i^* \oplus c^*}_{\text{Constant}}] = 2^{-k} .$$

Thus, $\Pr_{sk}[c^* \oplus sk = m_0^* | c^* \oplus sk = m_0^* \vee c^* \oplus sk = m_1^*] = \frac{1}{2}$. Thus the best strategy for even an omnipotent adversary is to output a random $b^* \leftarrow \{0,1\}$. Thus $\Pr[\text{Adversary wins}] = \frac{1}{2}$. □

Dealing With A Priori Information

- In addition, we also want to avoid the next type of attacks:
  - Adversary has seen an encryption of "Yes" before. Or
  - She has seen an encryption of a plaintexts beginning with "Yes". Or
  - She has seen a number of encryptions, of plaintexts, possibly all chosen by herself
- Formalised by allowing the adversary to access encryption oracles
- In particular, cryptosystem has to be randomised

Secret-Key Cryptosystems: Syntactic Definition (2)

Definition

Secret-key cryptosystem is a triple of three algorithms:

- Key generation $G(k)$ that outputs a k-bit random string
- Encryption $E : \mathcal{K} \times \mathcal{M} \times \mathcal{R} \rightarrow \mathcal{C}$
- Decryption $D : \mathcal{K} \times \mathcal{C} \times \mathcal{R} \rightarrow \mathcal{M}$

We require additionally that $D_{sk}(E_{sk}(m; r)) = m$ for every sk, m, r.

Secret-Key Cryptosystems: IND-CPA Security

IND-CPA (indistinguishable against chosen plaintext attacks) **game**:

1. Challenger generates random key $sk \leftarrow G(k)$
2. **Query phase 1:**
   - For $i = 1$ to γ do:
     - Adversary $\mathcal{A}$ sends to challenger query m_i
     - Challenger replies with $c_i \leftarrow E_{sk}(m_i; r_i)$ for fresh random r_i
3. **Challenge phase:**
   - $\mathcal{A}$ chooses two messages (m_0^*, m_1^*) of equal length, and sends them to challenger
   - Challenger chooses random bit $b \leftarrow \{0,1\}$ and fresh random string r^*. She sends $c^* \leftarrow E_{sk}(m_b^*; r^*)$ to adversary
4. **Query phase 2:** As query phase 1, but for γ_2 queries
5. **Guessing phase:** $\mathcal{A}$ outputs bit b^*. She wins if $b^* = b$

Secret-Key Cryptosystems: IND-CPA Security

Definition

We say that a secret-key cryptosystem is $(\tau, \varepsilon, \mu_1, \gamma_1, \mu_2, \gamma_2)$-IND-CPA secure if $\Pr[\text{Adversary wins}] < \frac{1}{2} + \varepsilon$ for any adversary $\mathcal{A}$ that works in time τ and makes up to γ_i queries in phase i, and messages in query phase having max total length μ_i.

Goal: IND, quantitative: $(\tau, \varepsilon, \mu_1, \gamma_1, \mu_2, \gamma_2)$, qualitative: CPA.

Secret-Key Cryptosystems: Insufficiency of IND-CPA

- There are situation where security against CPA is insufficient
- In many situations, one can force receiver to decrypt ciphertext of her choice, and observe output
- For example
  - IPSec, where all packets have fixed-length header
  - Adversary can try to change few bytes of ciphertext, corresponding to packet number/sender
  - If modified ciphertext does not encrypt valid plaintext, receiver returns error
  - Adversary may get extra information
- Another example: auctions, where given encrypted bid $E_{\mathsf{sk}}(x)$ you may be able to compute $E_{\mathsf{sk}}(x + 1)$

IND-CCA Security

IND-CCA (indistinguishable against chosen ciphertext attacks) **game**:

1. Challenger generates random key $\mathsf{sk} \leftarrow G(k)$
2. **Query phase 1:**
   - $\mathcal{A}$ has adaptive access to encryption and decryption oracles
3. **Challenge phase:**
   - $\mathcal{A}$ chooses two messages (m_0^*, m_1^*) of equal length, and sends them to challenger
   - Challenger chooses random bit $b \leftarrow \{0, 1\}$ and fresh random string r^*. She sends $c^* \leftarrow E_{\mathsf{sk}}(m_b^*; r^*)$ to adversary
4. **Query phase 2:**
   - $\mathcal{A}$ has adaptive access to encryption and decryption oracles
   - ...except she is not allow to query $D_{\mathsf{sk}}(\cdot)$ on input c^*!
5. **Guessing phase:** $\mathcal{A}$ outputs bit b^*. She wins if $b^* = b$

Secret-Key Cryptosystems: IND-CCA Security

Definition

We say that a secret-key cryptosystem is $(\tau, \varepsilon, \mu_1, \gamma_1, \mu_2, \gamma_2)$-IND-CCA secure if $\Pr[\text{Adversary wins}] < \frac{1}{2} + \varepsilon$ for any adversary $\mathcal{A}$ that works in time τ and makes up to γ_i queries in phase i, and messages in query phase having max total length μ_i.

Goal: IND, quantitative: $(\tau, \varepsilon, \mu_1, \gamma_1, \mu_2, \gamma_2)$, qualitative: CCA.

Mystical World of Security Definitions

- There are many security definitions even for SKCs
  - Goal: IND (indistinguishability), NM (nonmalleability),...
  - Abilities: CPA, CCA1, CCA2 (=CCA), ...
- Some security definitions, though looking different, result in the same notion
- Example: IND-CCA2=NM-CCA2 [Bellare et al., 1998]
- We will discuss how to construct IND-CPA/IND-CCA secure SKCs in a later lecture

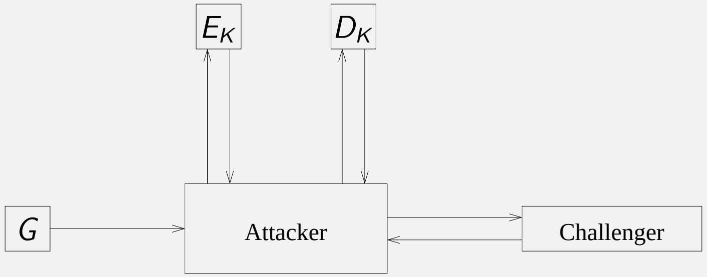
Mystical World of Security Definitions

It is extremely important to choose definition that suits your real-life situation. You must understand the situation to choose definition. You must understand definition to choose it

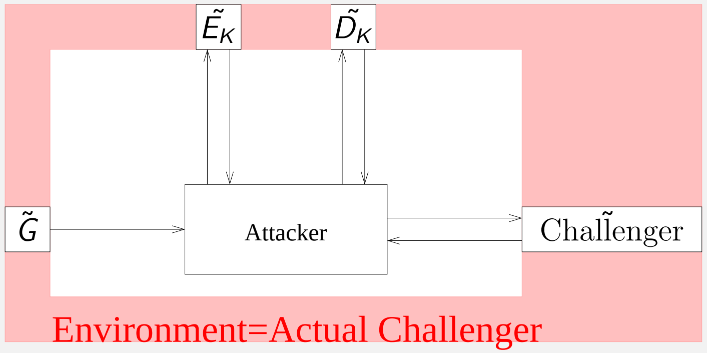
Plus: given concrete definition, you must be able to prove your solution is secure.

(Arguably, the most important lesson from this course.)

Game-Based Security: View Based On Environment



Game-Based Security: View Based On Environment



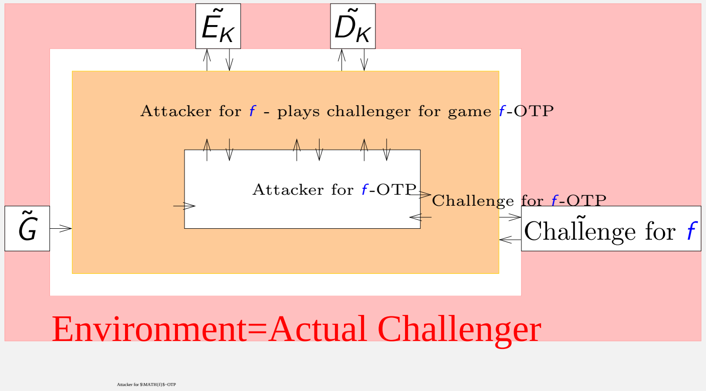
Reduction

- One-time pad is one of very few unconditionally secure protocols
  - There's unconditional authentication, secret sharing schemes, ...
  - In the case of encryption, OTP = most efficient unconditionally secure scheme
- Security of 99.9999% cryptographic primitives is conditional

Reduction

- **Typical theorem.** Assume X is (τ_X, ε_X)-secure in game GX. Then Y is (τ_Y, ε_Y)-secure in game GY.
  - If X is (τ_X, ε_X) GX-secure then Y is (τ_Y, ε_Y) GY-secure
  - If Y is not (τ_Y, ε_Y) GY-secure then X is not (τ_X, ε_X) GY-secure
  - Given adversary $\mathcal{A}_\mathcal{X}$ who (τ_X, ε_X) GX-breaks X, we construct adversary $\mathcal{A}_\mathcal{Y}$ who (τ_Y, ε_Y) GY-breaks Y
  - Adversary $\mathcal{A}_\mathcal{Y}$ simulates game GX to $\mathcal{A}_\mathcal{X}$, by inputting some values to $\mathcal{A}_\mathcal{X}$ and observing the outputs
  - Because $\mathcal{A}_\mathcal{Y}$ executes $\mathcal{A}_\mathcal{X}$, then trivially $\tau_Y \geq \tau_X$, and usually $\varepsilon_Y \leq \varepsilon_X$

Reductions: Sandbox view


$\tilde{E}_K$
$\tilde{D}_K$
Attacker for f - plays challenger for game f-OTP
Attacker for f-OTP
Challenge for f-OTP
$\tilde{G}$
Challenge for f
Environment=Actual Challenger
Attacker for $\MATH{f}$-OTP

Lecture 2

Some Simple Probability Facts

Fact 1. $\Pr[A = C] \leq \Pr[A = B] + \Pr[B = C]$.
Let q be the input domain
Proof:

$$\begin{aligned}\Pr[A = C] =& \Pr[A = C | A = B] \Pr[A = B] + \Pr[A = C | A \neq B] \Pr[A \neq B]\\ =& \Pr[B = C | A = B] \Pr[A = B] +\\ & \Pr[A = C | A \neq B, B = C] \Pr[A \neq b, B = C] +\\ & \Pr[A = C | A \neq B, B \neq C] \Pr[A \neq B, B \neq C]\\ \leq& \Pr[B = C] + \Pr[A = B | A \neq B, B = C] + 0\\ \leq& \Pr[B = C] + \Pr[A = B] \ .\end{aligned}$$

Some Simple Probability Facts

Fact 2. If $\Pr[A | \overline{F}] = \Pr[B | \overline{F}]$ then $| \Pr[A] - \Pr[B] \leq \Pr[F]$.
Proof:
Assume $\Pr[A] \geq \Pr[B]$. Now,

$$\begin{aligned}\Pr[A] - \Pr[B] \Pr[A|F] \Pr[F] + \Pr[A|\overline{F}] \Pr[\overline{F}] - \Pr[B|F] \Pr[F] - \Pr[B|\overline{F}] \Pr[\overline{F}]\\ = \Pr[A|F] \Pr[F] - \Pr[B|F] \Pr[F] \leq \Pr[A|F] \Pr[F] \leq \Pr[A|F] \ .\end{aligned}$$

By Bayes's theorem, $\Pr[A|F] = \Pr[F|A] \Pr[A] / \Pr[F]$. Thus

$$\begin{aligned}\ldots =& \Pr[F|A] \Pr[A] / \Pr[F]\\ \leq& \Pr[F|A] \Pr[A] \leq \Pr[F|A] \Pr[A] + \Pr[F|\overline{A}] \Pr[\overline{A}] = \Pr[F] \ .\end{aligned}$$

Intuition: If two things are similar unless something bad happens, they can be indistinguished only when this bad happens.

Computational Indistinguishability

Fix distributions $\mathcal{D}_1, \mathcal{D}_2$ as public parameters. Consider game IND-KPA:

1. **Challenge phase:** Challenger picks random $b \leftarrow \{0, 1\}$. He sends $x \leftarrow D_b$ to adversary
2. **Guessing phase:** Adversary outputs bit b^*
3. Adversary wins if $b^* = b$

Definition

Two distributions $\mathcal{D}_1, \mathcal{D}_2$ are (τ, ε)-indistinguishable if for any τ-time adversary $\mathcal{A}$, probability that $\mathcal{A}$ wins is $\leq \frac{1}{2} + \varepsilon$.

Computational Indistinguishability: Example

Definition

Let U_n be random distribution on n-bit strings.

Definition

A function $f : \{0, 1\}^k \to \{0, 1\}^\ell$ such that $f(U_k), U_\ell$ are (τ, ε)-indistinguishable is (τ, ε)-key derivation function.

Statistical Difference

Definition

$\mathcal{D}_1, \mathcal{D}_2$ are ε-indistinguishable if they are (∞, ε)-indistinguishable. ε is statistical difference of two distributions.

Let S be set of input values on which some algorithm outputs 1, then

$$\begin{aligned}\varepsilon =& \frac{1}{2} \cdot \max_S |\Pr[\mathcal{D}_1 \in S] - \Pr[\mathcal{D}_2 \in S]| \\ =& \frac{1}{2} \sum_x |\Pr[\mathcal{D}_1 = x] - \Pr[\mathcal{D}_2 = x]| \enspace .\end{aligned}$$

Clearly, if $\mathcal{D}_1, \mathcal{D}_2$ are ε-indistinguishable then they are (τ, ε)-indistinguishable for any τ.

Example: f-OTP

One-time pad is defined as follows:

- $G(k)$ returns a k-bit string.
- $E : \{0,1\}^k \times \{0,1\}^k \rightarrow \{0,1\}^k$ returns $E_{\mathsf{sk}}(m) := \mathsf{sk} \oplus m$.
- $D : \{0,1\}^k \times \{0,1\}^k \rightarrow \{0,1\}^k$ returns $D_{\mathsf{sk}}(c) := \mathsf{sk} \oplus c$.

Good: OTP is $(\infty, 0)$-IND-KPA-secure.
Bad: one-time key is as long as message.
Define f-OTP for key derivation function $f : \{0,1\}^k \rightarrow \{0,1\}^\ell$, $\ell \geq k$:

- $G(k)$ returns a k-bit string.
- $E : \{0,1\}^k \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$ returns $E_{\mathsf{sk}}(m) := f(\mathsf{sk}) \oplus m$.
- $D : \{0,1\}^k \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$ returns $D_{\mathsf{sk}}(c) := f(\mathsf{sk}) \oplus c$.

Security Proofs By Game Chains

- Security of complex protocols depends usually on many conditional statements.
- Common technique: game-hopping = a chain of several games, every game takes care of one statement

Security Proofs By Game Chains

- We want to prove that protocol is $\mathbf{Game_0}$-secure
- It is often difficult to do it directly, so instead:
- One defines an environment/new challenger who may interact differently with adversary
- Let this be $\mathbf{Game_1}$
- Let W_i be the event that $\mathcal{A}$ wins in $\mathbf{Game_i}$
- Let D_i be the event that $\mathcal{A}$ can distinguish games $\mathbf{Game_i}$ and $\mathbf{Game_{i+1}}$
- Clearly $\Pr[W_0] \leq \Pr[W_1] + \Pr[D_0]$
- **Game chains:** Often one defines chain of games $\mathbf{Game_1}, \dots, \mathbf{Game_m}$, then $W_0 \leq W_m + \sum D_i$
- After that, one upperbounds values W_m and D_i
- See the next lecture!

Simple Probability Fact: $\Pr[W_0] \le \Pr[D_0] + \Pr[W_1]$

Let W_i be event that adversary wins in game $\mathbf{Game_i}$, D_i be event that some adversary distinguishes games $\mathbf{Game_i}$ and $\mathbf{Game_{i+1}}$. Why $\Pr[W_0] \le \Pr[D_1] + \Pr[W_1]$? Define "bad" event F_i, such that: games $\mathbf{Game_i}$ and $\mathbf{Game_{i+1}}$ differ only if F_i holds. Then:

$$\begin{aligned}\Pr[W_0] =& \Pr[W_0|\overline{F}_0]\Pr[\overline{F}_0] + \Pr[W_0|F_0]\Pr[F_0]\\ =& \Pr[W_1|\overline{F}_0]\Pr[\overline{F}_0] + \Pr[W_0|F_0]\Pr[F_0]\\ \le& \Pr[W_1] + \Pr[F_0] \enspace .\end{aligned}$$

But D_0 is F_0 in eyes of some efficient adversary. Thus if $\Pr[D_0]$ is probability that any efficient adversary can distinguish games then $\Pr[W_0] \le \Pr[W_1] + \Pr[D_0]$.
By induction, we can also prove that $\Pr[W_0] \le \Pr[W_m] + \Pr[D_0] + \Pr[D_1] + \cdots + \Pr[D_{m-1}]$.

f-OTP Is Computationally IND-KPA Secure

- Define two games $\mathbf{Game_0}$, $\mathbf{Game_1}$
- In $\mathbf{Game_0}$, challenger creates a random k-bit key sk, receives ℓ-bit messages m_0^*, m_1^* from adversary, returns $c^* := m_b^* \oplus f(\mathsf{sk})$ for a random bit $b \leftarrow \{0,1\}$
- This is the original game, we have to show that W_0 is small
- In $\mathbf{Game_1}$, challenger creates a random ℓ-bit key sk', receives ℓ-bit messages m_0^*, m_1^* from adversary, returns $c^\dagger := m_b^* \oplus \mathsf{sk}'$ for a random bit $b \leftarrow \{0,1\}$
- $\mathbf{Game_1}$ corresponds to OTP, thus $W_1 = 0$ for arbitrary adversary
- Thus $W_0 \le D_1$, where D_1 is probability adversary distinguishes games $\mathbf{Game_0}$, $\mathbf{Game_1}$

f-OTP Is Computationally IND-KPA Secure (Continued)

- Equal to probability adversary distinguishes c^* and $c^\dagger$ for unknown b
- Because sk' is random, $c^\dagger$ is random
- Thus W_0 is upperbounded by probability that adversary distinguishes c^* from random
- Because $c^* := m_b^* \oplus f(\mathsf{sk})$, This is upperbounded by probability that adversary distinguishes $f(\mathsf{sk})$ from random

f-OTP Is Computationally IND-KPA Secure: Formal Statement

The previous argument was informal. But following it, we can prove next theorem.

Theorem

If $f : \{0,1\}^k \rightarrow \{0,1\}^\ell$ is a (τ, ε)-key derivation function then f-OTP is $(\tau - small\ value, \varepsilon)$-IND-KPA secure.

f-OTP Is Computationally IND-KPA Secure: Proof

Proof.

We only have to bound D_1. Assume $\mathcal{A}$ can (τ^*, ε)-distinguisher between games $\mathbf{Game_0}$ and $\mathbf{Game_1}$. Construct next $\mathcal{B}$ that distinguishes output of f from random.

$\mathcal{B}$ gets ℓ-bit input c, where c is random if $b = 0$ and $c = f(x)$ for random k-bit string x if $b = 1$.

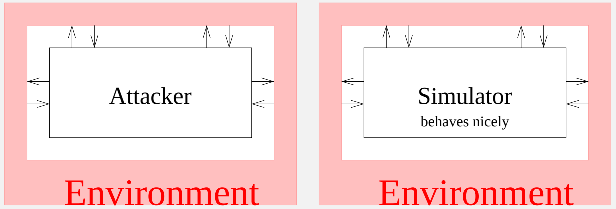
$\mathcal{B}$ receives m^* from $\mathcal{A}$. She gives $m^* \oplus c$ as challenge to $\mathcal{A}$ who outputs $b^* = 0$ if she thinks she is in $\mathbf{Game_0}$, and $b^* = 1$ if she thinks she is in $\mathbf{Game_1}$.

$\mathcal{B}$ outputs b^*. Clearly, if $\mathcal{A}$ guessed correctly then also $\mathcal{B}$ guesses correctly. Moreover, $\mathcal{B}$'s execution time is only slightly larger than $\mathcal{A}$'s time. □

Comments On f-OTP

- Good: key is shorter than messages.
- Bad: key can still only be used once.
- Somewhat more complex construction is backbone of CTR-mode and many modern streamciphers.
- There, one can use single key to encrypt many messages
- . . . under slightly more complex assumptions
- See a later lecture.

Simulation-Based Security



For every attacker, there exists nicely behaving simulator, such that environment does not distinguish whether he is communicating with attacker or simulator
(We may talk about it during last lectures)

Public-Key Cryptosystems: Syntactic Definition

- All participants of encryption (Alice, Bobs, . . .) have often access to some common public parameters
- Every party has secret key sk and public key pk.
- If Alice sends a message to Bob, she first obtains Bob's public key pk, encrypts by using that
- Bob decrypts the ciphertext using his secret key sk

Public-Key Cryptosystems: Syntactic Definition

Definition

Public-key cryptosystem is a triple of three algorithms:

- Key generation $G(k)$ that outputs secret/public keys $(\mathsf{sk}, \mathsf{pk})$
- Randomised encryption $E_{\mathsf{pk}}(m; Vr) = c$
- Deterministic decryption $D_{\mathsf{sk}}(c) = m$

We require additionally that $D_{\mathsf{sk}}(E_{\mathsf{pk}}(m; Vr)) = m$ for every m, Vr and every $(\mathsf{sk}, \mathsf{pk}) \in G(k)$.

We omit public parameters from most of the formal definitions, algorithms however use them implicitly

Hybrid And Usual PKCs

- SKC-s encrypt arbitrary bitstrings.
- Common PKC-s encrypt messages of some fixed size
  - RSA: elements modulo n
  - Many other schemes: elements of some finite group

  "Non-hybrid PKC" or just "'PKC"
- PKC-s that encrypt arbitrary bitstrings are called hybrid PKC-s.

Security Definitions For PKC

- Security definitions are similar to case of SKC
- One major difference:
  - Because everybody knows public key, there is no need to access encryption oracle!
- Plus of course, encryption is done by using public-key

Secret-Key Cryptosystems: IND-CPA Security

IND-CPA (indistinguishable against chosen plaintext attacks) **game**:

1. Challenger generates random key $\mathsf{sk} \leftarrow G(k)$
2. **Query phase 1:**
   - For $i = 1$ to γ_1 do:
     - Adversary $\mathcal{A}$ sends to challenger query m_i
     - Challenger replies with $c_i \leftarrow E_{\mathsf{sk}}(m_i; r_i)$ for fresh random r_i
3. **Challenge phase:**
   - $\mathcal{A}$ chooses two messages (m_0^*, m_1^*) of equal length, and sends them to challenger
   - Challenger chooses random bit $b \leftarrow \{0, 1\}$ and fresh random string r^*. She sends $c^* \leftarrow E_{\mathsf{sk}}(m_b^*; r^*)$ to adversary
4. **Query phase 2:** As query phase 1, but for γ_2 queries
5. **Guessing phase:** $\mathcal{A}$ outputs bit b^*. She wins if $b^* = b$

Public-Key Cryptosystems: IND-CPA Security

No need for encryption oracles — adversary can encrypt herself

IND-CPA (indistinguishable against chosen plaintext attacks) **game**:

1. Challenger generates random key $(\mathsf{sk}, \mathsf{pk}) \leftarrow G(k)$
2. **Challenge phase:**
   - $\mathcal{A}$ chooses two messages (m_0^*, m_1^*) of equal length, and sends them to challenger
   - Challenger chooses random bit $b \leftarrow \{0,1\}$ and fresh random string r^*. She sends $c^* \leftarrow E_{\mathsf{pk}}(m_b^*; r^*)$ to adversary
3. **Guessing phase:** $\mathcal{A}$ outputs bit b^*. She wins if $b^* = b$

In fact, for PKC, IND-KPA = IND-CPA!

Public-Key Cryptosystems: IND-CPA Security

Definition

We say that a public-key cryptosystem is (τ, ε)-IND-CPA secure if $\Pr[\text{Adversary wins}] < \frac{1}{2} + \varepsilon$ for any adversary $\mathcal{A}$ that works in time τ.

Goal: IND, quantitative: (τ, ε), qualitative: CPA.

Reminder: Groups

- Assume $\mathbb{G}$ is a group with operation $\cdot$
  - $a(bc) = (ab)c$ for any $a, b, c \in \mathbb{G}$— associativity
  - There exists $1 \in \mathbb{G}$ such that $1a = a1 = a$ for any $a \in \mathbb{G}$ — unit element
  - For any $a \in \mathbb{G}$ there exists $a^{-1} \in \mathbb{G}$ s.t. $aa^{-1} = a^{-1}a = 1$ — inverse
- If additionally $ab = ba$ for any $a, b \in \mathbb{G}$ then $\mathbb{G}$ is Abelian group
- Let $\mathrm{ord}(\mathbb{G}) := |\mathbb{G}|$ — order of group = number of elements

Reminder: cyclic groups

- Denote $q := \mathrm{ord}(\mathbb{G})$ and $\mathbb{Z}_q := \{0, 1, \ldots, q-1\}$
- $\mathbb{G}$ is cyclic if: exists a generator $g \in \mathbb{G}$ such that $\mathbb{G} = \{g^0, g^1, g^2, \ldots, g^{q-1}\}$
  - For any $h \in \mathbb{G}$ there exists unique $i \in \mathbb{Z}_q$ such that $h = g^i$
- Write $i = \log_g h$ — i is discrete logarithm of h on basis g
- Fact: groups of prime order do not have nontrivial subgroups
  - Follows from Lagrange theorem

In following, we always assume $\mathbb{G}$ is has prime order, is cyclic, Abelian.

Reminder: Diffie-Hellman Key Exchange

- Alice, Bob have secret keys sk_a, sk_b and public keys pk_a, pk_b
- They want to generate a common secret key
- Idea: they share a cyclic finite group $\mathbb{G}$ with generator g
- Public keys are defined as $\mathsf{pk}_a = g^{\mathsf{sk}_a}$, $\mathsf{pk}_b = g^{\mathsf{sk}_b}$
- Alice computes common secret key as $\mathsf{pk}_b^{\mathsf{sk}_a} = (g^{\mathsf{sk}_b})^{\mathsf{sk}_a} = g^{\mathsf{sk}_a \mathsf{sk}_b}$
- Bob computes common secret key as $\mathsf{pk}_a^{\mathsf{sk}_b} = (g^{\mathsf{sk}_a})^{\mathsf{sk}_b} = g^{\mathsf{sk}_a \mathsf{sk}_b}$
- They use then $g^{\mathsf{sk}_a \mathsf{sk}_b}$ for symmetric or public-key encryption

Note: this protocol by itself is weak against meet-in-middle attacks but we are not going to elaborate on that, nor give security definitions for key exchange. See [Shoup, 1999] for definitions if interested

Discrete Logarithm Assumption

DL game:

1. Public parameters: $\mathbb{G}, g, q$
2. **Setup phase:** challenger generates random $\mathsf{sk} \leftarrow \mathbb{Z}_q$, sends $h := g^{\mathsf{sk}}$ to adversary
3. **Challenge phase:** adversary returns $x \in \mathbb{Z}_q$
4. Adversary wins if $h = g^x$

Definition

Group $\mathbb{G}$ is a (τ, ε)-DL group if for any τ-times adversary the probability that adversary wins is $\leq \frac{1}{q} + \varepsilon$

Example Groups

Those groups *should* be known from Crypto I.

- Fix 1024-bit prime p and 160-bit prime q, such that $q \mid p - 1$. Then $\mathbb{Z}_p^*$ has a unique subgroup of order q. Defined $\mathbb{G}$ to be this subgroup.
- A well-chosen elliptic curve group of prime size $q \approx 2^{160}$

In both cases, it is assumed that $\tau \approx 2^{80}$, $\varepsilon \approx 2^{-80}$

Fact (Crypto I). DL in any group of size $\approx 2^{2k}$ can be computed in $\approx 2^k$ steps by using several standard methods. (Standard cycle-finding methods: baby-step-giant step etc)

Elgamal Cryptosystem

- Public parameters: group $\mathbb{G}$ with generator g and size $q \approx 2^{2k}$
- Key generation $G(k)$: choose random $\mathsf{sk} \leftarrow \mathbb{Z}_q$. Set $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$
- Encryption E of a message $m \in \mathbb{G}$: generate random $r \leftarrow \mathbb{Z}_q$, set $E_{\mathsf{pk}}(m; r) = (c_1, c_2) := (m \cdot \mathsf{pk}^r, g^r)$
- Decryption D of a ciphertext $(c_1, c_2) \in \mathbb{G}^2$: compute $m := c_1 / c_2^{\mathsf{sk}}$
- Correctness: $c_1 / c_2^{\mathsf{sk}} = m \cdot \mathsf{pk}^r / (g^r)^{\mathsf{sk}} = m (g^{\mathsf{sk}})^r / (g^{\mathsf{sk}})^r = m$

Proposed in [Elgamal, 1985].

Elgamal = DH Key Exchange + OTP

Elgamal can be seen as OTP, with key computed by using DH key exchange

- r is Alice's temporary secret key, g^r is Alice's temporary public key, $\mathsf{pk}^r = g^{\mathsf{sk} \cdot r}$ is common secret key.
- Encryption = Alice's one-time public key, plus mK, with K = common one-time secret key
- We defined OTP by using XOR, but multiplication in cyclic group is as valid

(OTP can use any group, and any group operation)

IND-CPA Security Of Elgamal: Assumptions

- IND-CPA security of Elgamal follows from hardness of computing the "Diffie-Hellman" common secret key, given only both public keys
  - CDH assumption — computational Diffie-Hellman assumption
- More precisely, because we need indistinguishability, common secret key has to be indistinguishable from random
  - As in case of f-OTP, where secret key was $f(x)$, we had to assume output of f is indistinguishable from random

  DDH assumption — Decisional Diffie-Hellman assumption

IND-CPA Security Of Elgamal: Assumptions

- Unfortunately, in a general group DDH assumption is stronger than DL assumption
  - There exist groups were DDH assumption is wrong, but both DL and CDH assumptions are believed to be true
- Thus one assumes DDH explicitly in security proof. Also CDH is usually a separate assumption

CDH Assumption

CDH game:

1. Public parameters: $\mathbb{G}, g, q$
2. **Challenge phase:** Challenger generates random $x, y \leftarrow \mathbb{Z}_q$. He sends g^x, g^y to adversary. Adversary returns $h \in \mathbb{G}$
3. Adversary wins if $h = g^{xy}$

Definition

Group $\mathbb{G}$ is a (τ, ε)-CDH group if for any τ-times adversary the probability that adversary wins is $\leq \frac{1}{q} + \varepsilon$

For simplicity, we define $DH(g^x, g^y) := g^{xy}$.

DDH Assumption

DDH game:

1. Public parameters: $\mathbb{G}, g, q$
2. **Challenge phase:** Challenger generates random $x, y, z \leftarrow \mathbb{Z}_q$ and $b \leftarrow \{0, 1\}$. Challenger sets $h_1 := g^x, h_2 := g^y$. If $b = 0$ then $h_3 := g^z$ else $h_3 \leftarrow g^{xy}$. He sends h_1, h_2, h_3 to adversary. Adversary returns $b^* \in \{0, 1\}$
3. Adversary wins if $b^* = b$

Definition

Group $\mathbb{G}$ is a (τ, ε)-DDH group if for any τ-times adversary the probability that adversary wins is $\leq \frac{1}{2} + \varepsilon$

DDH Tuples

Definition

We say (g, h_1, h_2, h_3) is DDH tuple if $(h_1, h_2, h_3) = (g^x, g^y, g^{xy})$ for some x, y.

Thus in DDH game, adversary has to distinguish random DDH tuples from random group elements

Home Problems

Problem

Show that if $\mathbb{G}$ is a (τ, ε)-CDH group then $\mathbb{G}$ is a $(\tau - \textit{small value}, \varepsilon)$-DL group.

Problem

Show that if $\mathbb{G}$ is a (τ, ε)-DDH group then $\mathbb{G}$ is a $(\tau - \textit{small value}, \varepsilon)$-CDH group.

IND-CPA Security of Elgamal

Theorem

Assume $\mathbb{G}$ is a (τ, ε)-DDH group. Then Elgamal is $(\tau - \textit{small value}, \varepsilon/2)$-IND-CPA secure.

In proof, common parameters: $(\mathbb{G}, g, q)$ — known by everybody.

Proof of IND-CPA Security of Elgamal (1/5)

Assume τ^*-time $\mathcal{A}$ can break Elgamal with probability ε.
Construct adversary $\mathcal{B}$ that tries to break DDH.
Challenger generates random $b_\beta \leftarrow \{0,1\}$, and challenge (h_1, h_2, h_3) as required by game: if $b_\beta = 0$ then h_3 is random, otherwise $h_3 = DH(h_1, h_2)$. $\mathcal{B}$ has to guess b_β.
Intuitive idea: $\mathcal{B}$ plays challenger for $\mathcal{A}$, receives a guess b_α^* from $\mathcal{A}$ and then outputs his guess b_β^*.

Proof of IND-CPA Security of Elgamal (2/5)

Simulation of IND-CPA game between $\mathcal{B}$ and $\mathcal{A}$:

- **Setup phase:** $\mathcal{B}$ lets $\mathsf{pk}_\beta := h_1$, sends h_1 to $\mathcal{A}$. He does not know secret key but it does not matter because $\mathcal{A}$ does not see it, and public key has correct distribution.
- **Challenge phase:** After receiving (m_0^*, m_1^*) from $\mathcal{A}$, $\mathcal{B}$ sets $c^* = (c_0^*, c_1^*) := (m_\gamma^* \cdot h_3, h_2)$ for random $\gamma \leftarrow \{0,1\}$. He sends c^* as challenge to $\mathcal{A}$, and gets back $\mathcal{A}$'s guess b_α^*.

After this game $\mathcal{B}$ guesses $b_\beta^* := 1$ if $\mathcal{A}$ guesses γ correctly, and $b_\beta^* = 0$ otherwise.

Proof of IND-CPA Security of Elgamal (3/5)

Simulation of IND-CPA game between $\mathcal{B}$ and $\mathcal{A}$:

- **Setup phase:** $\mathcal{B}$ lets $\mathsf{pk}_\beta := h_1$, sends h_1 to $\mathcal{A}$. He does not know secret key but it does not matter because $\mathcal{A}$ does not see it, and public key has correct distribution.
- **Challenge phase:** After receiving (m_0^*, m_1^*) from $\mathcal{A}$, $\mathcal{B}$ sets $c^* = (c_0^*, c_1^*) := (m_\gamma^* \cdot h_3, h_2)$ for random $\gamma \leftarrow \{0,1\}$. He sends c^* as challenge to $\mathcal{A}$, and gets back $\mathcal{A}$'s guess b_α^*.

If $b_\beta = 1$ then $c_0^* = m_\gamma^* DH(h_1, h_2)$ is correct encryption of m_γ^*.
Thus $\mathcal{A}$ guesses γ correctly with probability $\frac{1}{2} + \varepsilon$ and $\mathcal{B}$ returns $b_\beta^* = 1$ with probability $\Pr[b_\beta^* = 1 | b_\beta = 1] = \frac{1}{2} + \varepsilon$.

Proof of IND-CPA Security of Elgamal (4/5)

Simulation of IND-CPA game between $\mathcal{B}$ and $\mathcal{A}$:

- **Setup phase:** $\mathcal{B}$ lets $\mathsf{pk}_\beta := h_1$, sends h_1 to $\mathcal{A}$. He does not know secret key but it does not matter because $\mathcal{A}$ does not see it, and public key has correct distribution.
- **Challenge phase:** After receiving (m_0^*, m_1^*) from $\mathcal{A}$, $\mathcal{B}$ sets $c^* = (c_0^*, c_1^*) := (m_\gamma^* \cdot h_3, h_2)$ for random $\gamma \leftarrow \{0,1\}$. He sends c^* as challenge to $\mathcal{A}$, and gets back $\mathcal{A}$'s guess b_α^*.

If $b_\beta = 0$ then distribution of c^* is independent of γ. Thus $\mathcal{A}$ guesses γ correctly is $\frac{1}{2}$, and $\mathcal{B}$ returns $b_\beta^* = 1$ with probability $\Pr[b_\beta^* = 1 | b_\beta = 0] = \frac{1}{2}$.

Proof of IND-CPA Security of Elgamal (5/5)

But then probability $\mathcal{B}$ guesses correctly is

$$\begin{aligned}
\Pr[b^*_\beta = b_\beta] =& \Pr[b^*_\beta = b_\beta | b_\beta = 1] \Pr[b_\beta = 1] + \Pr[b^*_\beta = b_\beta | b_\beta = 0] \Pr[b_\beta = 0] \\
=& \frac{1}{2} \cdot \Pr[b^*_\beta = 1 | b_\beta = 1] + \frac{1}{2} \cdot \Pr[b^*_\beta = 0 | b_\beta = 0] \\
=& \frac{1}{2} \cdot \Pr[b^*_\beta = 1 | b_\beta = 1] + \frac{1}{2} \cdot (1 - \Pr[b^*_\beta = 1 | b_\beta = 0]) \\
=& \frac{1}{2} \cdot (\frac{1}{2} + \varepsilon) + \frac{1}{2} \cdot (1 - \frac{1}{2}) = \frac{1}{2} + \varepsilon/2
\end{aligned}$$

.

Thus $\Pr[\mathcal{B} \text{ wins}] = \varepsilon/2$.
Moreover, $\mathcal{B}$ only executes $\mathcal{A}$ once, computes one multiplication in group and does some other small computation. QED

If You Want, Read...

Course notes at `http://www-cse.ucsd.edu/users/mihir/cse207/` are good for this lecture (though we do not follow them) — see pseudorandom functions, symmetric encryption.

What We Know?

- How to define security by using games
- How to prove security by using reductions
  - Sandboxing: given $\mathcal{A}_\mathcal{X}$ who breaks game GX, construct an attacker $\mathcal{A}_\mathcal{Y}$ that breaks game GY
  - $\mathcal{A}_\mathcal{Y}$ plays challenger in game GX, and observers $\mathcal{A}_\mathcal{X}$'s responses in this game
- Definitions: IND-KPA, IND-CPA, IND-CCA
- For SKC: OTP is IND-KPA secure, if f if KDF then f-OTP is IND-KPA secure
- For PKC: IND-KPA = IND-CPA, if DDH is difficult then Elgamal is IND-CPA secure

Few Words About τ and ε

- τ and ε are dependent
  - Example: some problem may be $(2^{80}, 1)$-secure and $(2^{40}, 2^{-40})$ secure
  - If adversary has more time, she can break problem with higher probability
- There are conjectured τ, ε for which DL/CDH/DDH problems seem to be hard
- Exact reduction makes possible to specify how those values "carry over" when we construct some protocol
- In PKC, τ, ε depend on security parameter k — e.g., key size
  - Security of DDH depends on keys size $\Rightarrow$ security of Elgamal depends on key size
- In SKC, ciphers are usually designed for fixed k, so we think of k as being a constant

More About Computational Indistinguishability

In practice, adversary often sees several elements from a distribution.

Fix distributions $\mathcal{D}_1, \mathcal{D}_2$ as public parameters. Consider game:

1. **Challenge phase:** Challenger picks random $b \leftarrow \{0, 1\}$. He sends $x_1, \ldots, x_\mu \leftarrow \mathcal{D}_b$ to adversary
2. **Guessing phase:** Adversary outputs bit b^*
3. Adversary wins if $b^* = b$

Definition

Two distributions $\mathcal{D}_1, \mathcal{D}_2$ are (τ, ε, μ)-indistinguishable if for any τ-time adversary $\mathcal{A}$ that sees μ samples, probability that $\mathcal{A}$ wins is $\leq \frac{1}{2} + \varepsilon$.

Pseudorandom Generators

- Assume for $f : \{0, 1\}^k \to \{0, 1\}^\ell$, $f(U_k), U_\ell$ are (τ, ε)-indistinguishable.
- Then f is (τ, ε)-key derivation function.
- If $k > \ell$ then f is called (τ, ε)-pseudorandom generator, PRG.
- If $k \leq \ell$ then KDF's can be constructed without cryptographic assumptions.
- **Fact.** One can construct PRG given any one-way function. One needs one-way functions to construct PRG-s. [Impagliazzo et al., 1989]
- The corresponding reductions are inefficient, thus PRGs based on any OWFs are not used in practice.

Home Problems

Assume $f : \{0, 1\}^k \to \{0, 1\}^k \times \{0, 1\}^k$ is PRG, $f(x) = (y, z)$. Figure out if the next functions are PRG-s for reasonable quantitative parameters. (Prove or find attack.)

- $f_1(x) = (z, y)$
- $f_2(x) = (z, y, z \oplus y)$
- $f_3(x) = (0, x, y)$
- $f_4(x) = (f(x), f(y))$
- $f_5(x) = z$
- $f_6(x) = (x, y)$
- $f_7(x) = (x \oplus z, y)$

Reminder: f-OTP

Define f-OTP for PRG $f : \{0, 1\}^k \to \{0, 1\}^\ell$, $\ell \geq k$:

- $G(k)$ returns a k-bit string.
- $E : \{0, 1\}^k \times \{0, 1\}^\ell \to \{0, 1\}^\ell$ returns $E_{\mathsf{sk}}(m) := f(\mathsf{sk}) \oplus m$.
- $D : \{0, 1\}^k \times \{0, 1\}^\ell \to \{0, 1\}^\ell$ returns $D_{\mathsf{sk}}(c) := f(\mathsf{sk}) \oplus c$.

We proved that if $f : \{0, 1\}^k \to \{0, 1\}^\ell$ is a (τ, ε)-PRG then f-OTP is $(\tau - \text{small value}, \varepsilon)$-IND-KPA secure.

Secret-Key Cryptosystems: IND-CPA Security (reminder)

IND-CPA (indistinguishable against chosen plaintext attacks) **game**:

1. Challenger generates random key $\mathsf{sk} \leftarrow G(k)$
2. **Query phase 1:**
   - For $i = 1$ to γ do:
     - Adversary $\mathcal{A}$ sends to challenger query m_i
     - Challenger replies with $c_i \leftarrow E_{\mathsf{sk}}(m_i; r_i)$ for fresh random r_i
3. **Challenge phase:**
   - $\mathcal{A}$ chooses two messages (m_0^*, m_1^*) of equal length, and sends them to challenger
   - Challenger chooses random bit $b \leftarrow \{0, 1\}$ and fresh random string r^*. She sends $c^* \leftarrow E_{\mathsf{sk}}(m_b^*; r^*)$ to adversary
4. **Query phase 2:** As query phase 1, but for γ_2 queries
5. **Guessing phase:** $\mathcal{A}$ outputs bit b^*. She wins if $b^* = b$

IND-CPA Security Of Symmetric Cryptosystems

- IND-CPA allows adversary to make extra queries to encryption oracle
- f-OTP is IND-KPA but not IND-CPA secure
  - It was not randomized: for every K, m ciphertext was deterministic
  - Adversary can query encryption oracle with same m_0 she submits as challenge

Welcome To Counter Mode

- Execute f on K and on extra bitstring N
- We assume that knowing $f(K, N')$ for $N' \neq N$ gives no information on $f(K, N)$
- N is public, and shared by both guys who know secret key
- First possibility: N is required to be unique — then honest encrypter never uses same N with same key. We can then also assume adversary cannot query oracle on same N that was used while encryption
  - Because honest guys never uses the same N, adversary never sees two messages encrypted with same N, thus cannot get access to encryption oracle with same N

Welcome To Counter Mode

- Execute f on K and on extra bitstring N
- We assume that knowing $f(K, N')$ for $N' \neq N$ gives no information on $f(K, N)$
- N is public, and shared by both guys who know secret key
- Second possibility: N is always chosen randomly, and is unpredictable. Thus in query phase 1, adversary does not know N and thus can only guess it's values randomly. In query phase 2, she is forbidden to use the same N since that would break cryptosystem trivially

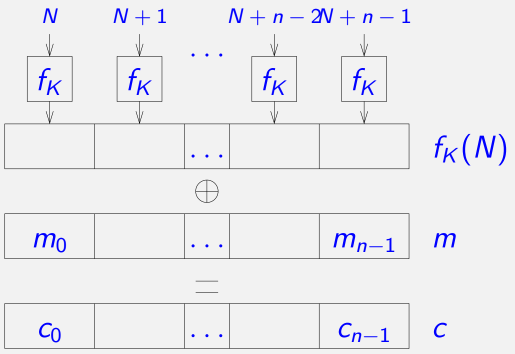
(Stateful) Counter Mode: Definition

- **Setup**: choose random sk, set $N \leftarrow 0$
- Assume $f : \mathcal{K} \times \{0,1\}^k \rightarrow \{0,1\}^k$
- Encryption (deterministic):
  - Let $m = m_0 || \dots || m_{n-1}$, where $|m_i| = k$
  - For $i = 0$ to $n-1$ do:
    - Set $c_i := f(K, N) \oplus m_i$
    - Set $N := N + 1$
  - Ciphertext is $c = (c_0, \dots, c_{n-1})$, store new value of N
- If N gets close to 2^k then generate new key

(Stateful) Counter Mode: Definition

- **Setup**: choose random sk, set $N \leftarrow 0$
- Assume $f : \mathcal{K} \times \{0,1\}^k \rightarrow \{0,1\}^k$
- Decryption (deterministic):
  - Let $c = c_0 || \dots || c_{n-1}$, where $|c_i| = k$
  - For $i = 0$ to $n-1$ do:
    - Set $m_i := f(K, N) \oplus c_i$
    - Set $N := N + 1$
  - Plaintext is $(m_0, \dots, m_{n-1})$, store new value of N

Counter Mode: Picture



Counter Mode: Statement Of Security

Theorem

If $f : \mathcal{K} \times \{0,1\}^k \rightarrow \{0,1\}^k$ is a (τ, ε)-pseudorandom permutation family and nonces are never reused, then stateful f-CTR is a $(\tau - O(\mu), \varepsilon + 0.5\mu^2/2^k, \gamma, \mu)$-IND-CPA secure symmetric cryptosystem.

There is no security guarantee if nonces are reused!
First proof in [Bellare et al., 1997].

Pseudorandom Function Families (Under CPA)

Assume $f : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^k$. The set $f_{\mathsf{sk}} : \{0,1\}^k \to \{0,1\}^k$ is small subset of all functions $\mathsf{Func} : \{0,1\}^k \to \{0,1\}^k$.

PRF Game (under CPA):

1. **Setup phase:** Challenger picks random $b \leftarrow \{0,1\}$. If $b = 0$ then he picks random function $g \leftarrow \mathsf{Func}$, otherwise he picks random $\mathsf{sk} \leftarrow \mathcal{K}$ and sets $g \leftarrow f_{\mathsf{sk}}$.
2. **Query phase:** Adversary makes up to γ queries m_i to challenger. Challenger responds with $g(m_i)$.
3. **Guessing phase:** Adversary outputs b^*. She wins if $b^* = b$.

In CCA game, adversary can also access decryption oracle. By default we always uses CPA game

Pseudorandom Function Family: Definition

Definition

We say that a function family $f : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^k$ is $(\tau, \varepsilon, \gamma)$-PRF if $\Pr[\text{Adversary wins}] < \frac{1}{2} + \varepsilon$ for any adversary $\mathcal{A}$ that works in time τ and sees γ samples.

Pseudorandom Permutation Families (Under CPA)

Assume $f : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^k$. The set $f_{\mathsf{sk}} : \{0,1\}^k \to \{0,1\}^k$ is small subset of all permutations $\mathsf{Perm} : \{0,1\}^k \to \{0,1\}^k$.

PRF Game (under CPA):

1. **Setup phase:** Challenger picks random $b \leftarrow \{0,1\}$. If $b = 0$ then he picks random function $g \leftarrow \mathsf{Perm}$, otherwise he picks random $\mathsf{sk} \leftarrow \mathcal{K}$ and sets $g \leftarrow f_{\mathsf{sk}}$.
2. **Query phase:** Adversary makes up to γ queries m_i to challenger. Challenger responds with $g(m_i)$.
3. **Guessing phase:** Adversary outputs b^*. She wins if $b^* = b$

In CCA game, adversary can also access decryption oracle. By default we always uses CPA game

Pseudorandom Permutation Family: Definition

Definition

We say that a permutation family $f : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^k$ is $(\tau, \varepsilon, \gamma)$-PRP if $\Pr[\text{Adversary wins}] < \frac{1}{2} + \varepsilon$ for any adversary $\mathcal{A}$ that works in time τ and sees γ samples.

How To Construct PRPs/PRFs?

- Again: PRP = PRF = OWF (one exists iff others exist) [Impagliazzo et al., 1989]
  - But known OWF's are much slower than blockciphers
  - + reductions are very loose and inefficient
- Assumption: AES is PRP
- In fact, PRPs were defined to model existing blockciphers and constructions, based on them
- Same constructions are often more efficient when based on PRFs, but there are no such symmetric primitives that directly work as PRFs
  - It is known how to efficiently construct PRFs from PRPs
  - . . . and PRPs from PRFs — Feistel's construction used already in DES, proven secure in [Luby and Rackoff, 1988]

How To Handle Random Functions?

- Assume $f : \{0,1\}^k \to \{0,1\}^\ell$ is random function
- This means that:
  - f is function: if we have seen $f(x)$ then the next time we see $f(x)$, it has same value
  - If we have not seen $f(x)$ then for us, $f(x)$ is totally random, that is, a priori probability that $\Pr[f(x) = y]$ is equal to $2^{-\ell}$ for any y

How To Handle Random Functions?

- One can construct a random function on the run by using that knowledge:
  - Create empty database
  - If $f(x)$ is queried then:
    - If (x, y) is in database for some y, return y
    - Otherwise, pick random $y \leftarrow \{0,1\}^\ell$, store (x, y) in database, return y
- Alternatively, one party can generate RF and then send its description — $2^k\ell$ bits — to partner
- Main motivation of using PRF-s is to get much shorter description without losing much in security

Fact: For any A, B there are $|B|^{|A|}$ functions $f : A \to B$

Random Oracle Model

- Random oracle = random function accessed only as oracle
- If security proof uses random oracle then proof is in ROM
- Sometimes, proof in ROM can be extended to case without random oracles — this is known as standard model
- However, there are protocols that are secure in ROM and not in standard model
- There are even tasks that can be solved ROM and that have no protocol whatsoever in standard model
- Proof in ROM can only be seen as argument that protocol is secure, it must always be followed by proof in standard model!

Random Function Is Pseudorandom Permutation (1/3)

Lemma (PRP-PRF Switching Lemma)

For any $\mu > 0$, Func is $(\infty, 0.5\mu^2/2^k, \mu)$-PRP.

Intuition.

For f drawn randomly from either Func or $\mathcal{PRP}$, the values $f(m)$ are random. Only difference is that in first case, $f(m)$ are completely random, while in second case, if $m \neq m'$ then $f(m) \neq f(m')$. Thus optimal distinguisher makes μ queries f at different locations and then returns 0 ("from Func") if all values are different, and 1 if two values are equal. □

While intuition is clear, full proof is not trivial — original proof (though not result) was found to be incorrect 5 years later.

Random Function Is Pseudorandom Permutation (2/3)

- Assume $\mathcal{A}$ is adversary playing the PRP vs PRF game
- Let F be bad event that not all answers returned by oracle are different
- Let F_i be bad event that ith query returns repetitive answer
- Clearly, adversary's view of both games are same until F happens
- Thus her winning probability is bound by $\Pr[F]$
- But $\Pr[F] \leq \sum \Pr[F_i]$

(This is essentially birthday paradox.)

Random Function Is Pseudorandom Permutation (3/3)

- Before ith queries, adversary has seen at most $i-1$ different values $f(x)$
- Thus $\Pr[F_i] \leq (i-1)/2^k$
- Thus $\Pr[F] \leq \sum_{i=1}^{\mu} (i-1)/2^k = \mu(\mu-1)/2^{k+1} \leq 0.5\mu^2/2^k$

Note: if $\mu \approx 2^{k/2}$ then $\varepsilon \approx 1$, thus one must have $\mu \ll 2^{k/2}$ to get any security.

(Reminder) Counter Mode: Statement Of Security

Theorem

If $f : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^k$ is a (τ, ε)-pseudorandom permutation family, and nonces are never reused, then stateful f-CTR is a $(\tau - O(\mu), \varepsilon + 0.5\mu^2/2^k, \gamma, \mu)$-IND-CPA secure symmetric cryptosystem.

There is no security guarantee if nonces are reused!
First proof in [Bellare et al., 1997].

Proof: CTR Mode Is Secure

- Proof has three games
- **Game$_0$** is original game
- **Game$_1$** replaces f with random permutation (CTR-mode with random permutation)
- **Game$_2$** replaces f with random function (CTR-mode with random function)

Proof: CTR Mode Is Secure

- We can bound $\Pr[D_0]$ by using assumption
- We can bound $\Pr[D_1]$ by using PRP-PRF switching lemma
- Thus we only need to bound $\Pr[W_2]$
- But if we never reuse nonce then in **Game$_2$**, message is XOR-d with completely random string, thus $\Pr[W_2] = 0$
- Thus $\Pr[W_0] \le \Pr[D_0] + \Pr[D_1] + \Pr[W_2] = \varepsilon + \mu(\mu - 1)/2^{k+1}$
- Computing of attacking time is trivial

Remarks On CTR Mode

- IND-CPA definition was modified — security is only guaranteed if nonce is nonrepeating
  - In practice, this means encrypter has to be stateful and memorise the last version of N
- We can only encrypt up to $2^{k/2}$ messages — up to 2^{64} with AES, up to 2^{32} with DES because of birthday paradox
- If we use PRF instead of PRP, we can encrypt 2^k messages — but there is no standard for PRF
- CTR mode is very efficient: $f_K(N + i)$ can be precomputed, thus after receiving message m, encryption = XOR
- CTR mode can be done in parallel: every ciphertext only depends on one plaintext
- No decryption f_K^{-1} — simpler implementations

Stateless CTR

- Like stateful CTR, except counter N is every time chosen randomly
- No need for state
- However, computing random numbers is expensive
- Security is slightly degraded because of that
  - In particular, CTR with random function suffers from birthday paradox

CTR Mode With Random Nonce: Security Statement

Theorem

If $f : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^k$ is a (τ, ε)-pseudorandom permutation family, then f-CTR is a $(\tau - O(\mu), \varepsilon + \mu^2/2^k, \gamma, \mu)$-IND-CPA secure symmetric cryptosystem.

CTR Mode With Random Nonce: Security Proof

- Follows the proof of stateful CTR
- Only difference: CTR mode with random function, $\Pr[W_2]$
- $\Pr[W_2]$ is bound by probability $\Pr[F]$ that two random N's are equal
- As before, $\Pr[F] \leq 0.5\mu^2/2^k$
- $\Pr[W_0] \leq \Pr[D_0] + \Pr[D_1] + \Pr[W_2] \leq \mu^2/2^k$

Note: real proof as in cited course notes is of course much longer

CBC Mode: History

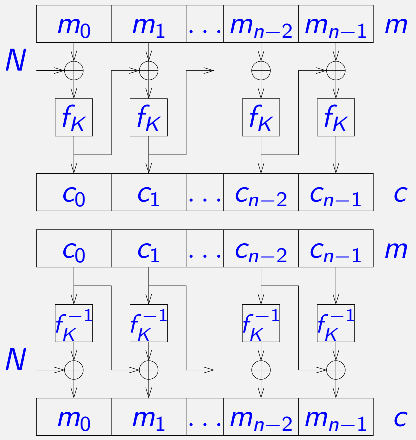
- CBC mode was one of four modes of operations for DES, standardised in 70s
  - CTR mode was not one of them
- In CBC, every subsequent ciphertext depends on all previous plaintexts
- No parallelisability
- Error propagation: if c_i gets modified then $m_{j \geq i}$ get modified
  - In CTR mode, if c_i is modified then only m_i is modified

  Currently this is said not to be cryptographic issue - use error-correcting codes

CBC Mode: Definition

- **Setup**: choose random sk, set $N \leftarrow 0$
- Assume $f : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^k$
- Encryption:
  - Choose random $N \leftarrow \{0,1\}^k$, set $N_0 := N$
  - Let $m = m_0 || \dots || m_{n-1}$, where $|m_i| = k$
  - For $i = 0$ to $n - 1$ do:
    - Set $c_i := f(K, m_i \oplus N)$
    - Set $N_{i+1} := c_i$
  - Ciphertext is $c = (N, c_0, \dots, c_{n-1})$

CBC Mode: Picture



CBC Mode: Security Statement

Theorem

If $f : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^k$ is a (τ, ε)-pseudorandom permutation family, then f-CBC is a $(\tau - O(\mu), \varepsilon + \mu^2/2^k, \gamma, \mu)$-IND-CPA secure symmetric cryptosystem.

CBC: Security Proof

- As in case of CTR mode, define three games
- **Game$_0$** — CBC with PRP
- **Game$_1$** — CBC with RP
- **Game$_2$** — CBC with RF
- As in CTR mode, we get $\Pr[W_0] \le \varepsilon + 0.5\mu^2/2^k + \Pr[W_2]$
- We only have to bound $\Pr[W_2]$

CBC Mode: Security Proof

- Consider CBC with random function
- Let F be event that some N has repeated, let F_i be event that N_i is first repeating nonce
- Clearly if the values $m_i \oplus N_i$ are all random and different then c is completely random — OTP
- Bound F_i:
  - For $j < i$, $c_j = N_j$ is new random, thus $f_K(m_j \oplus N_j)$ is completely random
  - Thus $N_i = c_{i-1}$ is completely random, and probability $\Pr[F_i]$ it is equal to some previous N_j is $\le (i-1)/2^k$
- Because every N_i is new and random then $f_K(N_i \oplus m_i)$ is new and random, thus under $\neg F$, CBC mode is secure
- Thus $\Pr[W_2] \le \Pr[F] \le \sum_{i=1}^{\mu} \Pr[F_i] \le 0,5\mu^2/2^k$

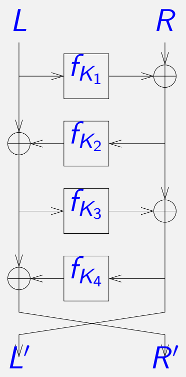
Motivation

- CTR+PRP: can only encrypt $2^{k/2}$ messages, birthday paradox
- CTR+PRF: no such problems
- But: AES is “standard” PRP family, no standard PRF families
- It would be desirable to construct efficient PRF families from PRP families

Motivation

- Existing block ciphers like DES use Feistel construction
- Gets us from round functions to permutation
- What can be said about security?
- Luby and Rackoff [Luby and Rackoff, 1988] answered affirmatively: with four rounds, Feistel construction transforms PRF to PRF

Luby-Rackoff: Picture



Luby-Rackoff: Formal Definition

- For functions $f_{K_i} : \{0,1\}^k \to \{0,1\}^k$, and parameter $r > 1$, define permutation $LR^f_K : \{0,1\}^{2k} \to \{0,1\}^{2k}$:
- Here, $K = (K_1, \dots, K_r)$, r independent keys
- For $m \in \{0,1\}^{2k}$, let $m = (L, R)$ for $L = L_0, R = R_0 \in \{0,1\}^k$
- For $i = 1$ to r do:
  - If i is odd then $L_i := L_{i-1}; R_i := R_{i-1} \oplus f_{K_i}(L_{i-1})$
  - If i is even then $R_i := R_{i-1}; L_i := L_{i-1} \oplus f_{K_i}(R_{i-1})$
- Set $c = (R_r, L_r)$
- Decryption is exactly the same with m, c interchanged

Luby-Rackoff: Remarks

- We are not going to prove security
- Reason 1: too long
- Reason 2: there are more efficient generic constructions

Problem

Show that LR is a permutation. Show that 3-round LR is not a PRP family.

Starting To Generalise

For $x \in \{0,1\}^{2k}$, denote $x = (x_L, x_R)$

Definition (Basic Feistel Permutation)

For function $f : \{0,1\}^k \to \{0,1\}^k$, let $\overline{f} : \{0,1\}^k to \{0,1\}^k$ be permutation defined as $\overline{f}(x) = (x_R, x_L \oplus f(x_R))$.

Definition (Feistel Network)

Let $f_1, \ldots, f_r$ be functions from $\{0,1\}^k$ to $\{0,1\}^k$. Denote by $LR(f_1, \ldots, f_r)$ permutation on $\{0,1\}^{2k}$ defined as

$$LR(f_1, \ldots, f_r) = \overline{f_r} \circ \cdots \circ \overline{f_1} \quad .$$

Security Statement: Luby-Rackoff

Theorem (Luby-Rackoff)

Let F be a PRF family. Then $\{LR(f_1, f_2, f_3, f_4) : f_i \in F\}$ is a PRP family.

Universal Hash Functions

Definition

Let $H = \{H_K\} : D \to \mathbb{G}$ be family of functions, where $\mathbb{G}$ is Abelian group. H is ε-almost strongly universal hash family if $\Pr[x \leftarrow \mathbb{G}, y \leftarrow \mathbb{G} \setminus \{x\} : h(x) = a, h(y) = b] \leq \varepsilon / |\mathbb{G}|$ for any $a, b \in \mathbb{G}$.
If $\varepsilon = 1/|\mathbb{G}|$ then H is strongly universal.

UH: Examples

- Let $H_{c,d} = cx + d \mod p$, where $c, d \in \mathbb{Z}_p$ and $c \neq 0$. Then $H : \mathbb{Z}_p \to \mathbb{Z}_p$ is strongly UH family.
- Really: for any $a, b \in \mathbb{Z}_p$, $\Pr_{x,y}[cx + d = a, cy + d = b] = 1/|\mathbb{G}|^2$
- Exactly one pair x, y: $x = (a - d)/c, y = (b - d)/c \mod p$

There exist more efficient UH families, but even this one shows we can construct one without any cryptographic assumptions.

XOR-Universal Hash Functions

Definition ([Krawczyk, 1994])

Let $H : D \to \{0,1\}^k$ be family of functions. H is ε-almost-xor-universal hash family if for any $x \neq y$ and any a, $\Pr[h \leftarrow H : h(x) \oplus h(y) = a] \leq \varepsilon$.

Known also as ε-AXU hash families.

AXU Hash Families: Examples

- Let p_i belong to finite field $GF(2^k)$
- Then polynomial $p(x) = \sum_{i=0}^{d} p_i x_i$ has d zeros
- For $\vec{p} = (p_d, \dots, p_0) \in GF(2^k))$ define $H_x(\vec{p}) = p(x)$, i.e., $H_x : GF(2^k)^{d+1} \to GF(2^k)$
- Let $H = \{H_x\} : x \in GF(2^k)$ with $H_x : GF(2^k)^{d+1} \to GF(2^k)$
- Because $H_x(\vec{p}) \oplus H_x(\vec{q}) \oplus a = p(x) \oplus q(x) \oplus a = (p \oplus q)(x) \oplus a$ is another polynomial, it has also at most d zeros
- Thus $\Pr[x \leftarrow GF(2^k) : H_{\vec{p}}(x) - H_{\vec{p}}(x) = a] \leq d/2^k$

Patel-Ramzan-Sundaram Construction

Theorem ([Patel et al., 1999])

Let h_1, h_2 be ε-AXU hash functions that are also permutations, and let f be random function. Then $LR(h_1, f, f, h_2)$ is $(\infty, O(\mu^2 - \varepsilon), \mu)$-PRP family.

We will not prove it. Note that as in any LR construction, key length of LR is somewhat longer than ideal. However, two outer rounds are much simpler than PRF, and two inner rounds may use same key. Thus PRS has three independent keys, while Luby-Rackoff has four.

From PRP to PRF

Theorem

Assume $P = P_k$ is PRP family. Then F, where $F_{k_1,k_2}(x) = P_{k_1}(x) \oplus P_{x_2}(x)$ is a PRF family.

We will not prove it again. See [Lucks, 2000].

General Idea

- When sending message over Internet, it is not only important to be sure message is secret, it is also important to know it comes from correct party
- Too many applications to even mention

MAC — Secret-Key Authentication

- Message was sent by one of possibly many parties who know secret key sk — it is assumed such parties trust each other to some extent
- Studied from 60s, unconditional MACs are possible
  - but with long keys
- Best MACs are often much more efficient than encryption
- IND-CPA cryptosystem + MAC = IND-CCA2 cryptosystem

Signature Schemes

- Every party has secret key, others know her public key. Signing by using secret key, verification by using public key.
- Diffie and Hellman [Diffie and Hellman, 1976] proposed idea but no implementations
- [Rivest et al., 1978] proposed signature scheme, based on RSA decryption
- Not secure according to contemporary definitions
- Contemporary secure signature schemes are about as efficient as secure public-key cryptosystems, but slightly more complicated

Security Definitions

- There exist many bad definitions
- In case of good definitions, there is a certain hierarchy
- Bad: after seeing tag, find secret key/message
- Correct: existential unforgeability
- After seeing MACs on messages, possibly chosen by yourself, come up with (m, tag), s.t. verification succeeds and tag has not been MAC oracle's answer on query m
- In actual definition, attacker can make $\gamma_v \geq 1$ forgery attempts and wins if she succeeds at least once
  - Reason: success probabilities depend nontrivially on γ_v

MAC: Syntax

- MAC consists of three algorithms:
- **Key generation G:** $G(k)$ usually picks random uniform key sk from $\{0,1\}^k$
- **Tag generation T:** $T_{\mathsf{sk}}(m)$ computes a tag for message m
- **Tag verification V:** $V_{\mathsf{sk}}(m, \mathsf{tag})$ returns accept or reject
- Of course, if $V_{\mathsf{sk}}(m, T_{\mathsf{sk}}(m)) = \mathsf{accept}$
- Note: T_{sk} can be randomised, or depend on unique nonce/counter

MAC: WUF-CMA Game

Let $\mathcal{M}$ be message space, $\mathcal{T}$ be tag space, $T_{\mathsf{sk}} : \mathcal{M} \to \mathcal{T}$.
WUF-CMA (weak unforgeability under chosen message attacks) game:

1. **Setup phase:** Challenger chooses random new key sk
2. **Query phase:** Adversary queries challenger with γ_s messages m_i. Challenger responds with $\mathsf{tag}_i := T_{\mathsf{sk}}(m_i)$.
3. **Challenge phase:** Adversary outputs γ_v forgery attempts $(m_j^*, \mathsf{tag}_j^*) \in \mathcal{M} \times \mathcal{T}$ s.t. $m_j^* \notin \{m_i\}$
4. Adversary wins if $V_{\mathsf{sk}}(m_j^*, \mathsf{tag}_j^*) = \mathsf{accept}$ for some j

MAC: SUF-CMA Game

Let $\mathcal{M}$ be message space, $\mathcal{T}$ be tag space, $T_{\mathsf{sk}} : \mathcal{M} \to \mathcal{T}$.
SUF-CMA (strong unforgeability under chosen message attacks) game:

1. **Setup phase:** Challenger chooses random new key sk
2. **Query phase:** Adversary queries challenger with γ_s messages m_i. Challenger responds with $\mathsf{tag}_i := T_{\mathsf{sk}}(m_i)$.
3. **Challenge phase:** Adversary outputs γ_v forgery attempts $(m_j^*, \mathsf{tag}_j^*) \in \mathcal{M} \times \mathcal{T}$, s.t. $(m_j^*, \mathsf{tag}_j^*) \notin \{(m_i, \mathsf{tag}_i)\}$
4. Adversary wins if $V_{\mathsf{sk}}(m_j^*, \mathsf{tag}_j^*) = \mathsf{accept}$ for some j

MAC: WUF/SUF-CMA Security Definition

Definition

MAC is $(\tau, \varepsilon, \gamma_s, \gamma_v)$-xUF-CMA secure if Pr[Adversary wins in xUF-CMA game] $\leq \varepsilon$ for any τ-time adversary that makes up to γ_s/γ_v queries to MAC/verification oracle

In general WUF-CMA is a stronger requirement (more about it when we talk about IND-CCA2-secure encryption).
Trivial adversary who outputs random tags wins with probability $1-(1-\frac{1}{|\mathcal{T}|})^{\gamma_v}$; we are interested in higher probabilities.

Every PRF Is MAC

Theorem

If $f : \{0,1\}^\ell \to \{0,1\}^k$ is $(\tau, \varepsilon, \gamma_s+\gamma_v)$-PRF then f is $(\tau - O(\gamma), 1-(1-\frac{1}{|\mathcal{T}|})^{\gamma_v}+\varepsilon, \gamma_s, \gamma_v)$-SUF-CMA secure MAC.

Every PRF Is MAC

- Let A be adversary who can break f as a MAC
- Construct B who breaks f as PRF
- B has an access to oracle g that is either PRF or RF
- If A makes MAC query m_i, B forwards it to oracle and returns $g(m_i)$
- Then B asks A for γ_v forgery attempts $(m_j^*, \mathsf{tag}_j^*)$. B queries g to check if A's forgery was successful. If it was, B guesses g is pseudorandom
- Security follows from fact that random function is ideally secure MAC: if g is RF then A $1-(1-\frac{1}{|\mathcal{T}|})^{\gamma_v}$-breaks g, otherwise A ε-breaks g

MACs with Nonces

As in case of symmetric cryptosystems, efficient MACs are easier to construct if they have nonces, $T_{\mathsf{sk}}(N, m) = \mathsf{tag}$ and $V_{\mathsf{sk}}(N, m, T_{\mathsf{sk}}(N, m)) = \mathsf{accept}$. As in case of CTR, we assume nonces are unique.
SUF-CMA game:

1. **Setup phase:** Challenger chooses random new key sk
2. **Query phase:** Adversary queries challenger with γ_s pairs N_i, m_i, s.t. $N_i \neq N_j$ for $i \neq j$. Challenger responds with $\mathsf{tag}_i := T_{\mathsf{sk}}(N_i, m_i)$.
3. **Challenge phase:** Adversary outputs γ_v forgery attempts $(N_j^*, m_j^*, \mathsf{tag}_j^*)$, s.t. $(N_j^*, m_j^*, \mathsf{tag}_j^*) \not\in \{(N_i, m_i, \mathsf{tag}_i)\}$
4. Adversary wins if $V_{\mathsf{sk}}(N_j^*, m_j^*, \mathsf{tag}_j^*) = \mathsf{accept}$ for some j

OTP

- PRF is MAC but also RF is trivially MAC
- OTP is MAC if key is used once
  - For ℓ-bit secret key sk, define $T_{\mathsf{sk}}(m) = \mathsf{sk} \oplus m$
- Problem 1: key is too long
- Problem 2: you can only use it once
- Solution to problem 1: “hash” message to shorter message
- Solution to problem 2: instead of XORing with random new string, XOR with PRF at new point

Wegman-Carter MAC with OTP (Solving Problem 1)

- Let $H = \{H_\kappa\} : \{0,1\}^\ell \to \{0,1\}^k$ be ε-AXU hash family
- For $(k + \log|H|)$-bit secret key (sk, κ), define $T_{\mathsf{sk},\kappa}(m) = \mathsf{sk} \oplus H_\kappa(m)$
- Verification: $V_{\mathsf{sk},\kappa}(m, c)$ returns accept iff $c = \mathsf{sk} \oplus H_\kappa(m)$

Theorem

If H is ε-AXU hash family then WC is $(\infty, \varepsilon, 1, 1)$-SUF-CMA secure.

$\Rightarrow$ Efficient one-time MAC without cryptographic assumptions.
Defined in [Wegman and Carter, 1981] for SU hash, redefined in [Krawczyk, 1994] for AXU hash.

Wegman-Carter + OTP: Security Proof

- 1 query, thus adversary sees pair (m, tag), s.t. $\mathsf{sk} \oplus H_\kappa(m) = \mathsf{tag}$ (and nothing else)
- Adversary has to construct (m^*, tag^*) for $m^* \neq m$, s.t. $\mathsf{sk} \oplus H_\kappa(m^*) = \mathsf{tag}^*$
- Given what she knows, this is equivalent to constructing (m^*, tag^*), s.t. $H_\kappa(m) \oplus H_\kappa(m^*) = \mathsf{tag} \oplus \mathsf{tag}^*$
- But for any $m^* \neq m$, this probability is upperbound by ε
- Because she has no additional information, ε is also upperbound on her success

Example

- Assume $k \mid \ell$
- Let H be the polynomial hash, $H_\kappa(m) = \sum_{i=0}^{n-1} m_i \kappa^i$ in $GF(2^k)$
- Select secret k-bit key κ
- For concrete message $m = (m_0, \dots, m_{n-1})$, $T_\kappa(m) := \mathsf{sk} \oplus \bigoplus_{i=0}^{n-1} m_i \kappa^i$
- Recall here $\varepsilon = (n-1)/2^k$
- One-time key κ can be short, it only influences the value ε
- Take say $k = 80$, then for even $n = 2^{40}$, this scheme is reasonably secure
- We can still only authenticate one (even if very long) message

Wegman-Carter MAC with f-OTP

- Let $H = \{H_\kappa\} : \{0,1\}^\ell \to \{0,1\}^\ell$ be AXU hash family
- Let $f : \{0,1\}^k \to \{0,1\}^k$ be PRP family
- For $(k + \log|H|)$-bit secret key (sk, κ), define $T_{\mathsf{sk},\kappa}(N, m) = (N, f_{\mathsf{sk}}(N) \oplus H_\kappa(m))$
- Note: sk and κ are both independent and random

Defined in [Shoup, 1996] who first proved security for PRF-family f, and for PRP-family f he added the distance between PRF and PRP.
Bernstein [Bernstein, 2005] gave a more precise proof that works directly for PRP-family. His proof assumes that f has a small "interpolation probability".

Bernstein's Proof: Basic Idea

- In WC+OTP, adversary saw one query and made one forgery attempt
- Because H is ε-AXU and $\Pr[\mathsf{sk} = c]$ is small for every c, we get simple upperbound
- If adversary sees more than one queries, we have to upperbound bound $\Pr[f_{\mathsf{sk}}(N_1) = x_1 \wedge \cdots \wedge f_{\mathsf{sk}}(N_\gamma) = x_\gamma \wedge f_{\mathsf{sk}}(N^*) = x^*]$
- This latter probability is exactly interpolation probability
- Security of WC follows directly (though via a technical proof) from that

Interpolation Assumption on f

Definition ([Bernstein, 2005])

For a function family $f : A \to B$ we say that f has γ-order interpolation probability $Int_\gamma(f) = \delta$ if

$$\Pr_{\mathsf{sk}}[(f_{\mathsf{sk}}(N_1), \ldots, f_{\mathsf{sk}}(N_\gamma)) = (x_1, \ldots, x_\gamma)] \le \delta$$

for all tuples $(x_1, \ldots, x_\gamma, N_1, \ldots, N_\gamma)$, s.t. $N_i \neq N_j$ for $i \neq j$.

Bernstein's Theorem

Theorem

Assume $\gamma < 2^k - 1$. Let H be an ε-AXU. Let f be a function family, s.t.

$$Int_\gamma(f) \le \frac{\delta}{2^{\gamma k}} \quad ,$$

and

$$Int_{\gamma+1}(f) \le \frac{\delta\varepsilon}{2^{\gamma k}} \quad .$$

Let $T_{\mathsf{sk},\kappa}(N, m) := (N, f_{\mathsf{sk}}(N) \oplus H_\kappa(m))$. Then MAC is $(\infty, \gamma_v \cdot \delta\varepsilon/2^{\gamma k}, \gamma, \gamma_v)$-SUF-CMA secure.

Note: in proof, we bound success probability of one forgery attempt, then multiply it with γ_v.

WC + f-OTP: Security Proof

- Attacker sees γ tuples $(N_i, m_i, \mathsf{tag}_i)$, s.t. $f_{\mathsf{sk}}(N_i) \oplus H_\kappa(m_i) = \mathsf{tag}_i$.
- She outputs $(N^*, m^*, \mathsf{tag}^*)$; wins if $f_{\mathsf{sk}}(N^*) \oplus H_\kappa(m^*) = \mathsf{tag}^*$
- Probability: over coin tosses of challenger and attacker
- Fix any $\mathsf{tag} = (\mathsf{tag}_1, \ldots, \mathsf{tag}_\gamma)$. We bound probability that $(N^*, m^*, \mathsf{tag}^*)$ is successful forgery for this concrete tuple.
- Probability: over choice of sk, κ and adversary's coin tosses.
- Let A be event that for this fixed tag, $N^* \not\in \{N_1, \ldots, N_\gamma\}$.
- We have to bound probability $\Pr[f_{\mathsf{sk}}(N^*) = H_\kappa(m^*) \oplus \mathsf{tag}^*, f_{\mathsf{sk}}(N_1) = H_\kappa(m_1) \oplus \mathsf{tag}_1, \ldots, f_{\mathsf{sk}}(N_\gamma) = H_\kappa(m_\gamma) \oplus \mathsf{tag}_\gamma]$
- This probability is upperbound by $Int_{\gamma+1}(f) \cdot \Pr[A] + Int_\gamma(f) \cdot \varepsilon \cdot \Pr[\neg A] = \delta\varepsilon/2^{\gamma k}$

What's This Interpolation Probability?

- If $f : A \rightarrow B$ is random function then $Int_\gamma(f) = 1/|B|^\gamma$

What's This Interpolation Probability?

- If $f : A \rightarrow B$ is random permutation then $Int_\gamma(f) \leq \frac{1}{|B|} \cdot \frac{1}{|B|-1} \cdot \cdots \cdot \frac{1}{|B|-(\gamma-1)}$
- But

$$\prod_{i=0}^{\gamma-1} (x-i) = \sqrt{\prod_{i=0}^{\gamma-1} (x-i)(x-(\gamma-1-i))}$$

$$\geq \sqrt{\left(x^2\left(1-\frac{\gamma-1}{x}\right)\right)^\gamma} = \sqrt{x^{2\gamma}(1-(\gamma-1)/x)^\gamma}$$

- Thus

$$Int_\gamma(f) \leq \sqrt{\frac{(1-(\gamma-1)/|B|)^{-\gamma}}{|B|^{2\gamma}}} = \frac{(1-(\gamma-1)/|B|)^{-\gamma/2}}{|B|^\gamma}$$

Bernstein's Theorem for RF

Theorem

Assume $\gamma < 2^k - 1$ and $\varepsilon \geq 1/2^k$. Let H be an ε-AXU. Let f be random function. Let $T_{\mathsf{sk},\kappa}(N, m) := (N, f_{\mathsf{sk}}(N) \oplus H_\kappa(m))$. Then MAC is $(\infty, \gamma_v \cdot \varepsilon, \gamma, \gamma_v)$-SUF-CMA secure.

Here, write $\delta = 1$. We know

$$Int_\gamma(f) = 1/2^{\gamma k} = \delta/2^{\gamma k}$$

and

$$Int_{\gamma+1}(f) = \frac{1}{2^{(\gamma+1)k}} \leq \frac{\varepsilon}{2^{\gamma k}} = \frac{\delta\varepsilon}{2^{\gamma k}}$$

because $\varepsilon \geq 1/2^k$

Bernstein's Theorem for RP

Theorem

Assume $\gamma < 2^k - 1$ and $\varepsilon \geq 1/2^k$. Let H be an ε-AXU. Let f be random permutation. Let $T_{\mathsf{sk},\kappa}(N, m) := (N, f_{\mathsf{sk}}(N) \oplus H_\kappa(m))$. Then MAC is $(\infty, \gamma_v \cdot (1 - \gamma/2^k)^{-(\gamma+1)/2}\varepsilon, \gamma, \gamma_v)$-SUF-CMA secure.

Define $\delta = (1 - \gamma/2^k)^{-(\gamma+1)/2}$.
Then

$$Int_\gamma(f) \leq \frac{(1 - (\gamma - 1)/2^k)^{-\gamma/2}}{2^{\gamma k}} \leq \frac{\delta}{2^{\gamma k}}$$

and

$$Int_{\gamma+1}(f) \leq \frac{(1 - \gamma/2^k)^{-(\gamma+1)/2}}{2^{(\gamma+1)k}} \leq \frac{\delta}{2^{(\gamma+1)k}} \leq \frac{\delta\varepsilon}{2^{\gamma k}}$$

Bernstein's Theorem For PRP

- Assume f is a $(\tau, \varepsilon_{prp}, \gamma + \gamma_v)$-PRP
- **Game$_0$**: WC + PRP, **Game$_1$**: WC + RP
- Then $\Pr[W_0] \leq \Pr[D_1] + \Pr[W_1] \leq \gamma_v \cdot (1 - \gamma/2^k)^{-(\gamma+1)/2}\varepsilon + \varepsilon_{prp}$

Remarks

- Efficiency: in case of RP, we had multiplicand $(1 - \gamma/2^k)^{-(\gamma+1)/2}$. This is "small" if $\gamma < 2^{k/2}$
  - If $\gamma = 2^{k/2}$, this is $(1 - 2^{-k/2})^{(-1-2^{-k/2})/2}$
  - But $(1 - x)^{(-1-x)/2} = 1 + x/2 + O(x^2)$ by series expansion
  - Thus for $\gamma = 2^{k/2}$, this multiplier is ≈ 1
- Bernstein's paper uses nonstandard notation, and is somewhat difficult to read
- However, it is very instructive and recommended
- In concrete proof, one keeps precise account of what the probability is taken over, on conditions, etc
- Current presentation was again simplified

Example: Concrete WC MAC

- Let f be AES, then $k = 128$. Assume message length divides by k
- Let H be the polynomial hash, $H_\kappa(m) = \sum_{i=0}^{n-1} m_i \kappa^i$
  - This assumes we fix a representation of $\{0, 1\}^k$ as a field $GF(2^{128})$, any representation is ok
  - Because AES uses $GF(2^{128})$ already, we can actually share arithmetics there
- Select secret k-bit keys sk, κ
- For concrete message $m = (m_0, \dots, m_{n-1})$ and nonce N, $T_{\mathsf{sk},\kappa}(N, m) := (N, AES_{\mathsf{sk}}(N) \oplus \bigoplus_{i=0}^{n-1} m_i \kappa^i)$
- This is very efficient: one AES call + computation of degree $n - 1$ polynomial over $GF(2^{128})$ (fast)
- Recall here $\varepsilon = (n - 1)/2^k$

References I

Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. (1997).
A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation.
In *38th Annual Symposium on Foundations of Computer Science*, pages 394–403, Miami Beach, Florida. IEEE Computer Society.

Bellare, M., Desai, A., Pointcheval, D., and Rogaway, P. (1998).
Relations among Notions of Security for Public-Key Encryption Schemes.
In Krawczyk, H., editor, *Advances in Cryptology — CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45, Santa Barbara, USA. Springer-Verlag.

Bernstein, D. J. (2005).
Stronger Security Bounds for Wegman-Carter-Shoup Authenticators.
In Cramer, R., editor, *Advances in Cryptology — EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 164–180, Aarhus, Denmark. Springer-Verlag.

Diffie, W. and Hellman, M. E. (1976).
New Directions in Cryptography.
IEEE Transactions on Information Theory, IT-22:644–654.

Elgamal, T. (1985).
A public key cryptosystem and a signature scheme based on discrete logarithms.
IEEE Transactions on Information Theory, 31(4):469–472.

References II

Impagliazzo, R., Levin, L. L., and Luby, M. (1989).
Pseudo-random function generation from one-way functions.
In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 12–24, Seattle, Washington.

Krawczyk, H. (1994).
LFSR-based Hashing and Authentication.
In Desmedt, Y. G., editor, *Advances in Cryptology—CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 129–139, Santa Barbara, USA. Springer-Verlag.

Luby, M. and Rackoff, C. (1988).
How to Construct Pseudorandom Permutations from Pseudorandom Functions.
SIAM Journal of Computing, 17(2):373–386.

Lucks, S. (2000).
The Sum of PRPs Is a Secure PRF.
In Preneel, B., editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 470–484, Bruges, Belgium. Springer-Verlag.

Patel, S., Ramzan, Z., and Sundaram, G. S. (1999).
Towards Making Luby-Rackoff Ciphers Optimal and Practical.
In Knudsen, L., editor, *Fast Software Encryption '99*, volume 1636 of *Lecture Notes in Computer Science*, pages 171–185, Rome, Italy. Springer-Verlag.

References III

Rivest, R. L., Shamir, A., and Adleman, L. M. (1978).
A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.
Communications of the ACM, 21(2):120–126.

Shannon, C. E. (1949).
Communication theory of secrecy systems.
Bell Sys. Tech. J., 28:657–715.

Shoup, V. (1996).
On Fast and Provably Secure Message Authentication Based on Universal Hashing.
In Koblitz, N., editor, *Advances in Cryptology—CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 313–328, Santa Barbara, California, USA. Springer-Verlag.

Shoup, V. (1999).
On Formal Models for Secure Key Exchange.
Technical Report 1999/012, IACR.
Available from
`http://eprint.iacr.org/1999/012/`.

Wegman, M. N. and Carter, L. (1981).
New Hash Functions and Their Use in Authentication and Set Equality.
jcss, 22(3):265–279.