Purdue CS555: Cryptography
Lecture 7 Scribe Notes

Instructor: Hanshen Xiao
Teaching Assistant: Justin He

Fall 2025

Recap from Previous Lecture

- Completion of Goldreich-Levin Theorem
- Stateless Encryption
- Introduction to Pseudorandom Functions (PRFs)

Topics Covered in This Lecture

1. Construction of PRFs from PRGs: The Goldreich-Goldwasser-Micali (GGM) Construction
2. Security proof for PRF-based encryption (many messages)
3. Applications of PRFs:
    - Friend-or-Foe Identification
    - Message Authentication Codes (MACs)
    - Encryption secure against active attacks
    - Negative results in learning theory

1 Review: Pseudorandom Functions

Definition 1 (Pseudorandom Function Family). *A collection of functions*

$$\mathcal{F}_\ell = \{f_k : \{0,1\}^\ell \to \{0,1\}^m\}_{k \in \{0,1\}^n}$$

consists of:

- *n: key length (security parameter)*
- *ℓ: input length*
- *m: output length*
- *All parameters are polynomial in n*
- *$|\mathcal{F}_\ell| \leq 2^n$ (singly exponential)*

Compare with all functions: $ALL_\ell = \{f : \{0,1\}^\ell \to \{0,1\}^m\}$ with $|ALL_\ell| = 2^{m \cdot 2^\ell}$ (doubly exponential).

Definition 2 (PRF Security). *$\mathcal{F}_\ell$ is pseudorandom if for all p.p.t. distinguishers D with oracle access, there exists negligible μ such that:*

$$\left|\Pr[f \leftarrow \mathcal{F}_\ell : D^f(1^n) = 1] - \Pr[f \leftarrow ALL_\ell : D^f(1^n) = 1]\right| \leq \mu(n)$$

1.1 PRF-Based Stateless Encryption

Let $f_k : \{0,1\}^\ell \to \{0,1\}^m$ be a PRF where 2^ℓ is super-polynomial.

- $\text{Gen}(1^n)$: Generate random n-bit key k
- $\text{Enc}(k, m)$: Pick random $x \leftarrow \{0,1\}^\ell$, output $c = (x, y = f_k(x) \oplus m)$
- $\text{Dec}(k, c = (x, y))$: Output $f_k(x) \oplus y$

Correctness: $\text{Dec}(k, \text{Enc}(k, m)) = f_k(x) \oplus (f_k(x) \oplus m) = m$.

2 Security of Secret-Key Encryption

2.1 Single Message Security

For all messages m_0, m_1 and all p.p.t. distinguishers D:

$$|\Pr[k \leftarrow \mathcal{K} : D(\text{Enc}(k, m_0)) = 1] - \Pr[k \leftarrow \mathcal{K} : D(\text{Enc}(k, m_1)) = 1]| \le \mu(n)$$

2.2 Many Message Security (IND-CPA)

Definition 3 (Multi-Message Security)**.** *For all p.p.t. distinguishers D with oracle access:*

$$\left|\Pr[k \leftarrow \mathcal{K} : D^{\mathit{Left}(\cdot,\cdot)}(1^n) = 1] - \Pr[k \leftarrow \mathcal{K} : D^{\mathit{Right}(\cdot,\cdot)}(1^n) = 1]\right| \le \mu(n)$$

where:

- *Left*(m_L, m_R)*: returns Enc*(k, m_L)
- *Right*(m_L, m_R)*: returns Enc*(k, m_R)

2.3 Security Proof via Hybrid Argument

Theorem: The PRF-based encryption scheme is IND-CPA secure.

Proof Sketch. We use a hybrid argument with 5 hybrids:

Hybrid 0: D gets access to Left oracle. $c = (x, y = f_k(x) \oplus m_L)$

Hybrid 1: Replace f_k by random function. $c = (x, y = r_x \oplus m_L)$ where r_x is uniformly random for each distinct x.

Indistinguishability: By PRF security, Hybrid 0 $\approx$ Hybrid 1.

Hybrid 2: Replace by truly random output. $c = (x, y = r_x)$ (ignore left messages entirely)

Indistinguishability: By birthday paradox, with high probability all x values are distinct. Since r_x is uniformly random and m_L is XORed with it, $r_x \oplus m_L$ has the same distribution as r_x (one-time pad).

Hybrid 3: Encrypt right messages with random function. $c = (x, y = r_x \oplus m_R)$

Indistinguishability: Same argument in reverse: r_x and $r_x \oplus m_R$ have the same distribution (by one-time pad).

Hybrid 4: Replace random function by f_k (Right oracle). $c = (x, y = f_k(x) \oplus m_R)$

Indistinguishability: By PRF security (symmetric to Hybrid 0 $\to$ Hybrid 1).

Since Hybrid 0 $\approx$ Hybrid 4, the scheme is secure. $\square$

3 The GGM Construction: PRG $\Rightarrow$ PRF

3.1 Motivation: Length Extension Revisited

Recall: Given PRG $G : \{0,1\}^n \to \{0,1\}^{n+1}$, we can extend to $G' : \{0,1\}^n \to \{0,1\}^{m(n)}$ for any polynomial m.

Construction: Write $G(s) = G_0(s) \| G_1(s)$ where $G_0(s)$ is 1 bit and $G_1(s)$ is n bits. Build a tree by repeatedly applying G.

Problem: Accessing the i-th output bit takes time $\approx i$, which is exponential when $i \approx 2^\ell$.

3.2 GGM Construction

Theorem 1 (Goldreich-Goldwasser-Micali). *Let G be a PRG. Then for every polynomials $\ell = \ell(n)$ and $m = m(n)$, there exists a PRF family*

$$\mathcal{F}_\ell = \{f_s : \{0,1\}^\ell \to \{0,1\}^m\}_{s\in\{0,1\}^n}$$

Construction: Write $G(s) = G_0(s) \| G_1(s)$ where both $G_0(s)$ and $G_1(s)$ are n bits each.
Build a complete binary tree of depth ℓ:

- Root is labeled with seed s
- Each node with label v has left child $G_0(v)$ and right child $G_1(v)$
- Each leaf corresponds to a string $x \in \{0,1\}^\ell$

Function Definition:

$$f_s(x_1 x_2 \cdots x_\ell) = G_{x_\ell}(G_{x_{\ell-1}}(\cdots G_{x_1}(s)\cdots))$$

Properties:

- f_s defines 2^ℓ pseudorandom values (one per leaf)
- The x-th value can be computed using ℓ evaluations of PRG G (following path from root to leaf x)
- Time complexity: $\mathrm{poly}(n) \cdot \ell = \mathrm{poly}(n)$ since $\ell = \mathrm{poly}(n)$

Output length: We focus on $m = n$. Can be adjusted by:

- Smaller output: truncation
- Larger output: PRG expansion from the leaf value

3.3 Security Proof for GGM

Lemma 1 (PRG Repetition Lemma). *Let G be a PRG. For every polynomial $L = L(n)$:*

$$(G(s_1), G(s_2), \ldots, G(s_L)) \approx (u_1, u_2, \ldots, u_L)$$

where s_i are independent random seeds and u_i are independent random strings.

Proof. By hybrid argument. If there exists a distinguisher with advantage ϵ, then there exists a distinguisher for G with advantage $\geq \epsilon/L$. □

Theorem 2. *The GGM construction yields a secure PRF.*

Proof. Assume for contradiction there exists p.p.t. distinguisher D and polynomial p such that:

$$\left|\Pr[f \leftarrow \mathcal{F}_\ell : D^f(1^n) = 1] - \Pr[f \leftarrow \mathrm{ALL}_\ell : D^f(1^n) = 1]\right| \geq \frac{1}{p(n)}$$

Hybrid Argument by Tree Levels:
Hybrid 0 (Pseudorandom World): D queries f_s for the GGM PRF built from seed s.

- Tree has root s and internal nodes computed via G
- Leaves are values $b_1, b_2, \ldots, b_{2^\ell}$

Hybrid 1: Replace level 1 nodes with random values.

- Root's children $s_0 = G_0(s)$ and $s_1 = G_1(s)$ are replaced by independent random values
- Rest of tree built from these random values using G

Indistinguishability: By PRG security (using lazy evaluation to answer queries efficiently).
Hybrid 2: Replace level 2 nodes with random values.

- All 4 nodes at depth 2 are independent random values
- Rest of tree built from these using G

Indistinguishability: By PRG repetition lemma and PRG security.
Hybrid i: Replace all nodes at depth i with independent random values.
Hybrid ℓ (Random World): All leaves are independent random values.

- This is exactly a random function from ALL_ℓ

Analysis:
Let $p_i = \Pr[D^{H_i}(1^n) = 1]$ where H_i denotes Hybrid i.
We know: $|p_0 - p_\ell| \geq \epsilon = \frac{1}{p(n)}$
By averaging, for some i: $|p_i - p_{i+1}| \geq \frac{\epsilon}{\ell}$
Key Observation - Lazy Evaluation:
To simulate Hybrid i, we don't need to generate the entire tree. When D queries x:

1. Follow path from root to leaf x
2. For nodes at depth $\leq i$: use stored random values
3. For nodes at depth $> i$: compute via G

This allows efficient simulation in polynomial time.
Reduction to PRG:
If $|p_i - p_{i+1}| \geq \frac{\epsilon}{\ell}$, then there is a distinguisher for PRG with advantage $\geq \frac{\epsilon}{q\ell}$ where q is the number of queries D makes.
The reduction works by using the distinguisher between Hybrid i and Hybrid $i+1$ to distinguish:

- $(G(s_0^i), G(s_1^i), \ldots)$ from (random strings)

By PRG repetition lemma, this contradicts PRG security. □

4 Applications of PRFs

4.1 Friend-or-Foe Identification

Setting: Pete needs to identify himself to a base station in the presence of an adversary who can listen to and modify communications.
Challenge-Response Protocol:

- Pete has ID number and PRF key s
- Base station sends random challenge r
- Pete responds with $(ID, f_s(r))$
- Base station verifies using its copy of f_s

Security Intuition:
Adversary can collect polynomially many pairs $(r_i, f_s(r_i))$ (potentially of her choosing).
To impersonate Pete, adversary must produce $f_s(r^*)$ for a fresh random challenge r^*.
This is hard by the unpredictability property of PRFs.

Lemma 2 (Unpredictability of PRFs)**.** *Let $f_s : \{0,1\}^\ell \to \{0,1\}^m$ be a PRF. Consider an adversary who obtains $f_s(x_1), \ldots, f_s(x_q)$ for polynomial $q = q(n)$.*
If she can predict $f_s(x^*)$ for $x^* \notin \{x_1, \ldots, x_q\}$ with probability $\frac{1}{2^m} + \frac{1}{poly(n)}$, then she breaks PRF security.

Proof Sketch. Build a distinguisher for the PRF:

- Query oracle on $x_1, \ldots, x_q$ and x^*
- If predictor's output matches $f(x^*)$, output "pseudorandom"

- Otherwise output "random"

For truly random function, prediction succeeds with probability $\frac{1}{2^m}$.
For PRF, prediction succeeds with probability $\frac{1}{2^m} + \frac{1}{\text{poly}(n)}$ by assumption.
This distinguishes with non-negligible advantage. $\square$

Requirements: Input and output lengths must be $\omega(\log n)$ to ensure security against guessing attacks.

5 Message Authentication Codes (MACs)

5.1 The Authentication Problem

Setting: Alice wants to send message m to Bob, but adversary can intercept, modify, or inject messages (man-in-the-middle attack).

Goal: Bob should be able to verify that message came from Alice and was not modified.
Naive Approach: Use encryption?
Problem: Encryption schemes are typically *malleable*.

- One-time pad: Given $c = m \oplus k$, adversary can create $c' = m' \oplus k$ by computing $c \oplus m \oplus m'$
- PRF-based encryption: Given $c = (r, f_k(r) \oplus m)$, adversary can create $(r, f_k(r) \oplus m')$

Key Insight: *Privacy and integrity are different goals.*

5.2 Definition of MACs

Definition 4 (Message Authentication Code)**.** *A MAC consists of three algorithms:*

- *Gen*(1^n)*: Produces key* $k \leftarrow \mathcal{K}$
- *MAC*(k, m)*: Outputs tag* t *(may be deterministic)*
- *Ver*(k, m, t)*: Outputs Accept or Reject*

Correctness: $\Pr[\text{Ver}(k, m, \text{MAC}(k, m)) = \text{Accept}] = 1$

5.3 Security Definitions

Power of Adversary:

- Can see many pairs $(m_i, \text{MAC}(k, m_i))$
- Has oracle access to $\text{MAC}(k, \cdot)$
- Can obtain tags for messages of choice

This is called a **Chosen Message Attack (CMA)**.
Security Levels:

- **Total break:** Adversary recovers key k
- **Universal break:** Adversary can generate valid tag for every message
- **Existential break:** Adversary can generate valid tag for *some* new message

Standard: We require security against existential forgery.

Definition 5 (EUF-CMA Security)**.** *A MAC is* ***Existentially Unforgeable under Chosen Message Attack*** *if for all p.p.t. adversaries A:*

$$\Pr[k \leftarrow \mathcal{K}; (m, t) \leftarrow A^{MAC(k,\cdot)}(1^n) : Ver(k, m, t) = 1 \wedge (m, t) \notin Q] = negl(n)$$

where Q is the set of query-response pairs (m_i, t_i) that A obtained.

5.4 PRF-Based MAC Construction

Construction:

- $\text{Gen}(1^n)$: Generate PRF key k
- $\text{MAC}(k, m)$: Output $f_k(m)$
- $\text{Ver}(k, m, t)$: Accept if $f_k(m) = t$, reject otherwise

Theorem 3. *If f_k is a secure PRF, then the above construction is EUF-CMA secure.*

Proof Sketch. By the unpredictability lemma for PRFs.

Adversary makes queries $m_1, \ldots, m_q$ and obtains $f_k(m_1), \ldots, f_k(m_q)$.

To forge, adversary must output (m^*, t^*) where $m^* \notin \{m_1, \ldots, m_q\}$ and $t^* = f_k(m^*)$.

This requires predicting $f_k(m^*)$ on a new input, which succeeds with probability at most $\frac{1}{2^m} + \text{negl}(n)$ by PRF security. □

5.5 Replay Attacks

Issue: Adversary can send an old valid (m, tag) at a later time.

Note: Our definition does not rule this out.

Solutions in Practice:

1. **Timestamps:** Include timestamp in message. Send $(m, T, \text{MAC}(k, m\|T))$ where T is current time.
2. **Sequence numbers:** Append counter to messages (requires stateful MAC). Send $(m, \text{seq}, \text{MAC}(k, m\|\text{seq}))$.

6 Encryption Secure Against Active Attacks

Problem: MACs provide integrity but not privacy. Encryption provides privacy but not integrity.

Solution: Combine both.

6.1 Encrypt-then-MAC

Construction: Use two independent keys k, k'.

Encryption:

1. Encrypt: $c = (x, f_k(x) \oplus m)$
2. Tag: $\text{tag} = f_{k'}(c)$
3. Send: (c, tag)

Decryption:

1. Verify: Check if $f_{k'}(c) = \text{tag}$
2. If invalid, output $\perp$
3. If valid, decrypt: $m = f_k(x) \oplus y$

Security: This provides both:

- **Privacy:** From PRF-based encryption (IND-CPA)
- **Integrity:** From MAC (EUF-CMA)

Together, these provide security against active adversaries who can both:

- Eavesdrop on ciphertexts (passive attack)
- Inject or modify ciphertexts (active attack)

7 Applications to Learning Theory

7.1 Negative Results

Theorem 4 (Kearns and Valiant 1994). *Assuming PRFs exist, there are hypothesis classes that cannot be learned by polynomial-time algorithms.*

Intuition:

Learning Theory/ML: Given labeled examples $(x_i, f(x_i))$ for unknown f, learn hypothesis $h \approx f$ that generalizes.

Cryptography (PRFs): Construct function families $\{f_k\}$ for which it is hard to predict f_k on new input even given query access.

Connection: If you can learn a PRF family in polynomial time, you can break its security (predict on new inputs). Therefore, PRFs give hard-to-learn function classes.

7.2 More Negative Results

PRFs have been used to show computational hardness of learning:

- Intersections of halfspaces
- Agnostic learning of halfspaces
- Various concept classes in computational learning theory

These results establish fundamental limits on what can be efficiently learned, assuming standard cryptographic assumptions.