CIS 5560

Cryptography
Lecture 9

Course website:
pratyushmishra.com/classes/cis-5560-s25/

Slides adapted from Dan Boneh and Vinod Vaikuntanathan
Announcements

• HW 3 due next Friday
• HW2 due tomorrow!
Recap of last lecture
Pseudorandom Functions

Collection of functions \( \mathcal{F}_\ell = \{ F_k : \{0,1\}^\ell \rightarrow \{0,1\}^m \}_{k \in \{0,1\}^n} \)
- indexed by a key \( k \)
- \( n \): key length, \( \ell \): input length, \( m \): output length.
- Independent parameters, all poly(sec-param) = poly\( (n) \)
- \#functions in \( \mathcal{F}_\ell \leq 2^n \) (singly exponential in \( n \))

\textbf{Gen}(1^n): Generate a random \( n \)-bit key \( k \).

\textbf{Eval}(k, x) is a poly-time algorithm that outputs \( F_k(x) \)
Security: Cannot distinguish from random function

\[ \left| \Pr [A^{f_k}(1^n) = 1 \mid k \leftarrow \{0,1\}^\ell] - \Pr [A^F(1^n) = 1 \mid F \leftarrow \text{Fns}] \right| \leq \text{negl}(n). \]
PRP/Block Cipher

A **block cipher** is a pair of efficient algs. \((E, D)\):

A canonical example:

1. **AES**: \(n=128\) bits, \(k = 128, 192, 256\) bits
2. **3DES**: \(n=64\) bits, \(k = 168\) bits (historical)
Goldreich-Goldwasser-Micali PRF

Construction: Let $G(s) = G_0(s) || G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both $n$ bits each.

Each path/leaf labeled by $x \in \{0,1\}^\ell$ corresponds to $f_s(x)$. 
Today’s Lecture

• Proof of security for MAC
• Short MAC $\rightarrow$ Long MACs
Goldreich-Goldwasser-Micali PRF

Construction: Let $G(s) = G_0(s) || G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both $n$ bits each.

Each path/leaf labeled by $x \in \{0,1\}^\ell$ corresponds to $f_s(x)$. 
Goldreich-Goldwasser-Micali PRF

Construction: Let $G(s) = G_0(s) \| G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both $n$ bits each.

The pseudorandom function family $\mathcal{F}_\ell$ is defined by a collection of functions $f_s$ where:

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

\* $f_s$ defines $2^\ell$ pseudorandom bits.

\* The $x^{th}$ bit can be computed using $\ell$ evaluations of the PRG $G$ (as opposed to $x \approx 2^\ell$ evaluations as before.)
GGM PRF: Proof of Security

By contradiction. Assume there is a ppt \( D \) and a poly function \( p \) s.t.

\[
\left| \Pr \left[ A_f^k(1^n) = 1 \mid k \leftarrow \{0,1\}^\ell \right] - \Pr \left[ A_F^F(1^n) = 1 \mid F \leftarrow \text{Fns} \right] \right| \geq \frac{1}{p(n)}.
\]

The pseudorandom world

\[
f \leftarrow \mathcal{F}_\ell
\]

\[
x \quad \Downarrow \quad f(x)
\]

Distinguisher D

0/1

The random world

\[
f \leftarrow \text{Fns}
\]

\[
x \quad \Downarrow \quad f(x)
\]

Distinguisher D

0/1
The pseudorandom world:  
Hybrid 0 

Problem:  
Hybrid argument on leaves doesn’t work. Why?
The pseudorandom world:
Hybrid 0

Key Idea:
Hybrid argument by levels of the tree
The pseudorandom world:

Hybrid 0

Hybrid 1

\[ s \]

\[ G_0(s) \]

\[ G_1(s) \]

\[ G_0(G_0(s)) \]

\[ G_1(G_0(s)) \]

\[ G_{x,\ell}(G_{x,\ell-1}(\ldots(s))) \]

\[ b_1 \ b_2 \ b_3 \ldots \ b_x \ldots b_{2^\ell} \]

\[ s_0 \]

\[ s_1 \]

\[ b_1 \ b_2 \ b_3 \ldots \ b_x \ldots b_{2^\ell} \]

\[ x \]

\[ f(x) \]

\[ D \]
Hybrid 1

\[ s_0 \text{ and } s_1 \text{ are random} \]

\[ G_1(G_0(s)) \]

\[ b_1 \ b_2 \ b_3 \ldots \ b_x \ldots \ b_2^\ell \]

Hybrid 2

\[ s_{00}, \ldots s_{11} \text{ are random} \]

\[ s_{00} \quad s_{01} \quad s_{10} \quad s_{11} \]

\[ b_1 \ b_2 \ b_3 \ldots \ b_x \ldots \ b_{2^\ell} \]
The random world:
Hybrid $\ell$

$\text{Hybrid } \ell$

\[
\begin{align*}
\mathbf{b}_1 & \quad \mathbf{b}_2 & \quad \mathbf{b}_3 & \quad \cdots & \quad \mathbf{b}_x & \quad \cdots & \quad \mathbf{b}_{2\ell} \\
\hline
\mathbf{O} & \quad \mathbf{O} & \quad \mathbf{O} & \quad \cdots & \quad \mathbf{O} & \quad \cdots & \quad \mathbf{O}
\end{align*}
\]

\[x \quad \downarrow \quad f(x) \quad \uparrow \]

\[D\]
Hybrid $i$

$S_{0i}, \ldots, S_{1i}$ are random

$b_1, b_2, b_3, \ldots, b_x, \ldots, b_{2^\ell}$

Q: Is the function in the hybrid efficiently computable?

A: Yes! Lazy Evaluation.
**GGM PRF**

Theorem: Let G be a PRG. Then, for every polynomials $\ell, m$, there exists a PRF family $\mathcal{F}_\ell = \{ f_s : \{0,1\}^\ell \to \{0,1\}^m \}_{s \in \{0,1\}^n}$.

Some nits:

- **Expensive**: $\ell$ invocations of a PRG.

- **Sequential**: bit-by-bit, $\ell$ sequential invocations of a PRG.

- **Loss in security reduction**: break PRF with advantage $\varepsilon \implies$ break PRG with advantage $\varepsilon / q \ell$, where $q$ is an arbitrary polynomial = #queries of the PRF distinguisher. Tighter reduction? Avoid the loss?
The authentication problem

This is known as a **man-in-the-middle attack**.
How can Bob check if the message is indeed from Alice?
The authentication problem

We want Alice to generate a tag for the message $m$ which is **hard to generate** without the secret key $k$. 

Can also alter/inject more messages!
Wait... Does encryption not solve this?

\[ m \rightarrow_{Enc(k, m)} Bob \]

Alice
\[ \text{key } k \]

Bob
\[ \text{Key } k \]
Wait... Does encryption not solve this?

One-time pad (and encryption schemes in general) are **malleable**.
Wait... Does encryption not solve this?

One-time pad (and encryption schemes in general) are **malleable**.

Privacy and Integrity are very **different goals**!
Message Authentication Codes (MACs)

A triple of algorithms (Gen, MAC, Ver):

- **Gen**: Produces a key $k \leftarrow \mathcal{K}$.
- **MAC**: Outputs a tag $t$ (may be deterministic).
- **Ver**: Outputs Accept or Reject.

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

**Security**: *Hard to forge.* Intuitively, it should be hard to come up with a new pair $(m', t')$ such that Ver accepts.
What is the power of the adversary?

- Can see many pairs \((m, \text{MAC}(k, m))\).
- Can access a MAC oracle \(\text{MAC}(k, \cdot)\)
  - Obtain tags for message of choice.

This is called a *chosen message attack (CMA)*.
Defining MAC Security

- **Total break**: The adversary should not be able to recover the key $k$.
- **Universal break**: The adversary can generate a valid tag for every message.
- **Existential break**: The adversary can generate a new valid tag $t$ for some message $m$.

We will require MACs to be secure against the existential break!!
EUF-CMA Security

Existentially Unforgeable against Chosen Message Attacks

Want: \( \Pr((m, t) \leftarrow A^{MAC(k, \cdot)}(1^n), \ Ver(k, m, t) = 1, (m, t) \notin Q)) = \text{negl}(n) \).

where \( Q \) is the set of queries \( \left\{ (m_i, t_i) \right\}_i \) that \( A \) makes.
Let $I = (S, V)$ be a MAC.

Suppose an attacker is able to find $m_0 \neq m_1$ such that

$$\text{MAC}(k, m_0) = \text{MAC}(k, m_1) \quad \text{for } \frac{1}{2} \text{ of the keys } k \text{ in } K$$

Can this MAC be secure?

Yes, the attacker cannot generate a valid tag for $m_0$ or $m_1$

No, this MAC can be broken using a chosen msg attack

It depends on the details of the MAC

$$\text{Adv}[A, I] = \frac{1}{2}$$
Let $I = (S, V)$ be a MAC.

Suppose $\text{MAC}(k, m)$ is always 5 bits long.

Can this MAC be secure?

No, an attacker can simply guess the tag for messages.

It depends on the details of the MAC.

Yes, the attacker cannot generate a valid tag for any message.

$$\text{Adv}[A, I] = \frac{1}{32}$$
Dealing with Replay Attacks

• The adversary could send an old valid \((m, \text{tag})\) at a later time.
  – In fact, our definition of security does not rule this out.

• In practice:
  – Append a time-stamp to the message. Eg. \((m, T, \text{MAC}(m, T))\) where \(T = 21\ \text{Sep 2022, 1:47pm}\).
  – Sequence numbers appended to the message (this requires the MAC algorithm to be stateful).