Lecture 7: Security of the GGM PRF*, and Authentication*

Notes by Yael Kalai

MIT - 6.5620
Lecture 7 (September 24, 2025)

> ***Warning:*** This document is a rough draft, so it may contain bugs. Please feel free to email me with corrections.

Recap

- Last class we defined the notion of a CPA secure encryption and the notion of a pseudorandom function (PRF).

  Definition 1 (Pseudorandom function)**.** A PRF $F = \{F_\lambda\}_{\lambda\in\mathbb{N}}$, where for every $\lambda \in \mathbb{N}$, $F_\lambda : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$, has the property that for every poly-size $\mathcal{A}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$,

  $$|\Pr_{k\leftarrow\mathcal{K}_\lambda}[\mathcal{A}^{F_\lambda(k,\cdot)}(1^\lambda) = 1] - \Pr_{R_\lambda}[\mathcal{A}^{R_\lambda(\cdot)}(1^\lambda) = 1]| \leq \mu(\lambda)$$

  where R_λ is truly random function $R_\lambda : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$. For concreteness, we think of $\mathcal{X} = \{0,1\}^n$ and $\mathcal{Y} = \{0,1\}$.

- We constructed a CPA secure encryption from any PRF.
- We constructed a PRF from any PRG (the GGM construction).

Today

- Prove security of the GGM construction.
- Define Message Authentication Codes (MAC)
- Construct a MAC from any PRF.

Recall the GGM PRF *Construction*

Suppose we are given a length doubling PRG

$$G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}.$$

Denote by

$$G(x) = (G_0(x)), G_1(x))$$

where for every $b \in \{0,1\}$

$$G_b : \{0,1\}^\lambda \to \{0,1\}^\lambda.$$

Similarly, for every $b_1, b_2 \in \{0,1\}$ denote by

$$G_{b_1,b_2} : \{0,1\}^\lambda \to \{0,1\}^\lambda$$

the function

$$G_{b_1,b_2}(x) = G_{b_2}(G_{b_1}(x)).$$

More generally, for every $b_1, \ldots, b_i \in \{0,1\}$ denote by

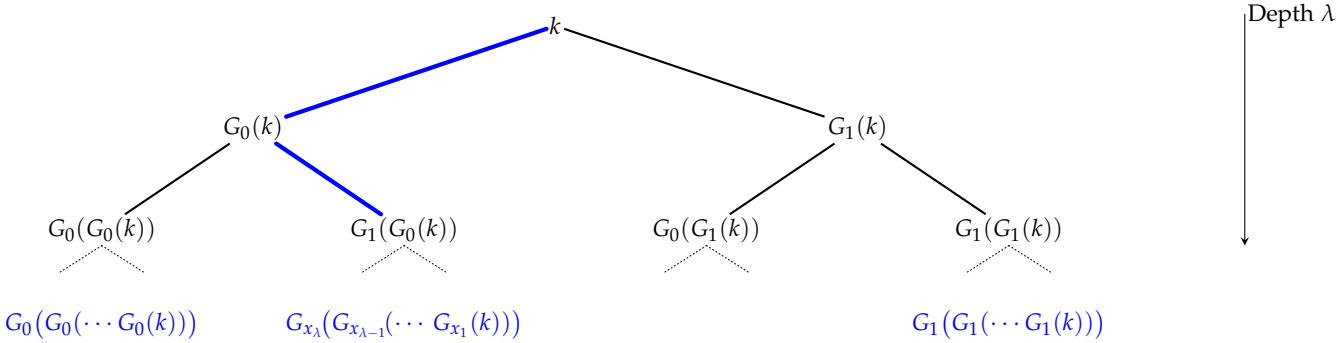
$$G_{b_1,\ldots,b_i} : \{0,1\}^\lambda \to \{0,1\}^\lambda$$

the function

$$G_{b_1,\ldots,b_i}(x) = G_{b_i}(\ldots(G_{b_1}(x))\ldots).$$

Goldreich–Goldwasser–Micali PRF

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



Each path/leaf labeled by $x \in \{0,1\}^\lambda$ corresponds to $F(k,x)$.

Theorem 2. *The GGM construction is a* PRF.

The proof of this theorem makes use of the following lemma.

Lemma 3. *If* $G : \{0,1\}^\lambda \to \{0,1\}^{n(\lambda)}$ *is a* PRG *then for every polynomial* $\ell : \mathbb{N} \to \mathbb{N}$ *it holds that*

$$\{G(k_1), \ldots, G(k_{\ell(\lambda)})\}_{\lambda \in \mathbb{N}} \approx \{U_{n(\lambda)\cdot\ell(\lambda)}\}_{\lambda \in \mathbb{N}}$$

where $k_1, \ldots, k_{\ell(\lambda)} \leftarrow \{0,1\}^\lambda$.

The proof of this lemma follows from a standard hybrid argument.

Proof of Lemma 3 Fix any poly-size $\mathcal{A}$. For every $\lambda \in \mathbb{N}$, and for every $i \in \{0, 1, \ldots, \ell(\lambda)\}$, denote by

$$H_{\lambda,i} = (G(k_1), \ldots, G(k_i), U_{n(\lambda)\cdot(\ell(\lambda)-i)}).$$

Then

$$\begin{aligned}
&|\Pr[\mathcal{A}(G(k_1), \ldots, G(k_{\ell(\lambda)}) = 1] - \Pr[\mathcal{A}(U_{n(\lambda)\cdot\ell(\lambda)}) = 1]| = \\
&|\Pr[\mathcal{A}(H_{\lambda,\ell}) = 1] - \Pr[\mathcal{A}(H_{\lambda,0}) = 1] = \\
&|\sum_{i=1}^{\ell} \Pr[\mathcal{A}(H_{\lambda,i}) = 1] - \Pr[\mathcal{A}(H_{\lambda,i-1}) = 1]| \leq \\
&\sum_{i=1}^{\ell} |\Pr[\mathcal{A}(H_i) = 1] - \Pr[\mathcal{A}(H_{i-1}) = 1]| \leq \\
&\mathrm{negl}(\lambda),
\end{aligned}$$

where the latter inequality follows from the fact that G is a PRG together with the fact that $\ell = \mathrm{poly}(\lambda)$. □

We next show how to use Lemma 3 to prove Theorem 2.

Proof of Theorem 2 The proof is via a hybrid argument on the layers of the tree. Suppose there exists a poly-size $\mathcal{A}$ and a non-negligible ϵ such that for every $\lambda \in \mathbb{N}$

$$|\Pr[\mathcal{A}^{F(k,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{R_\lambda}(1^\lambda) = 1]| \geq \epsilon$$

In what follows, we denote by $F_0 = F(k, \cdot)$. Let F_1 be the function that is similar to F_0 except that it replaces the first layer of the tree (from the root) with a uniform layer. Namely, $(G_0(k), G_1(k))$ is replaced with truly random (k_0, k_1). By the security of the PRG there exists a negligible function μ_1

$$|\Pr[\mathcal{A}^{F_0(k,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{F_1(k,\cdot)}(1^\lambda) = 1]| \leq \mu_1.$$

More generally, let F_i be the function that replaces the i'th layer of the tree (from the root) with truly random values. Note that $F_\lambda = R(\cdot)$. By a hybrid argument there exists $i \in [\lambda]$ such that for every $\lambda \in \mathbb{N}$

$$|\Pr[\mathcal{A}^{F_{i-1}}(1^\lambda) = 1] - \Pr[\mathcal{A}^{F_i}(1^\lambda) = 1]| \geq \frac{\epsilon}{\lambda}$$

We argue that this contradicts Lemma 3.

Note that the number of nodes in the ith layer is 2^i, which may be super-polynomial. Thus we cannot contradict Lemma 3 with $\ell = 2^i$ (indeed, this lemma is false with super-polynomial ℓ!). Instead, rather than assigning a string to each node in the i'th layer, we will only assign strings to nodes that are queried by $\mathcal{A}$.

Specifically, let $q = q(\lambda)$ be an upper bound on the number of oracle calls that $\mathcal{A}$ makes. We construct a poly-size adversary $\mathcal{B}$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr[\mathcal{B}(G(k_1), \ldots, G(k_q)) = 1] - \Pr[\mathcal{B}(U_{n(\lambda)\cdot 2q(\lambda)}) = 1]| \geq \frac{\epsilon}{\lambda}. \quad (1)$$

$\mathcal{B}$ on input $(x_{1,0}, x_{1,1}, \ldots, x_{q,0}, x_{q,1})$ emulates $\mathcal{A}(1^\lambda)$ as follows:

1. Upon receiving the j'th oracle call from $\mathcal{A}$ do the following:
   (a) Denote the oracle query by $r \in \{0,1\}^\lambda$.
   (b) Denote by $r_{[i-1]} \in \{0,1\}^{i-1}$ the first $i-1$ bits of r. Similarly denote by $r_{[i]} \in \{0,1\}^i$ the first i bits of r.
   (c) If $k_{r_{[i-1]},0}$ and $k_{r_{[i-1]},1}$ have not been previously defined, then set $k_{r_{[i-1]},0} = x_{j,0}$ and $k_{r_{[i-1]},1} = x_{j,1}$
2. Output $y = G_{r_\lambda}(G_{r_{\lambda-1}}(\ldots(k_{r_{[i]}})\ldots))$.

It is easy to see that if $(x_{1,0}, x_{1,1}, \ldots, x_{q,0}, x_{q,1})$ is uniformly distributed in $\{0,1\}^{\lambda\cdot 2q}$ then

$$\Pr[\mathcal{B}(x_{1,0}, x_{1,1}, \ldots, x_{q,0}, x_{q,1}) = 1] = \Pr[\mathcal{A}^{F_i}(1^\lambda) = 1]$$

On the other hand, if $(x_{1,0}, x_{1,1}, \ldots, x_{q,0}, x_{q,1})$ is distributed as $(G(k_1), \ldots, G(k_q))$ for random $k_1, \ldots, k_q \leftarrow \{0,1\}^\lambda$, then

$$\Pr[\mathcal{B}(x_{1,0}, x_{1,1}, \ldots, x_{q,0}, x_{q,1}) = 1] = \Pr[\mathcal{A}^{F_{i-1}}(1^\lambda) = 1].$$

Thus, by Equation (1), we conclude that

$$|\Pr[\mathcal{B}(G(k_1), \ldots, G(k_q)) = 1] - \Pr[\mathcal{B}(U_{\lambda\cdot 2q}) = 1]| \geq \frac{\epsilon}{\lambda}$$

contradicting Lemma 3. □

PRF *for Authentication*

So far we were focused on encryption, the task of communicating in a secret manner. We spent three weeks figuring out how to do this. We showed how to use a OWP to construct a PRG,[1] how to use a PRG to construct a PRF, and finally how to use a PRF to construct a CPA-secure encryption (i.e., a scheme that remains secure even if the adversary gets to see ciphertexts of any messages of its choice, which he can choose adaptively).

[1] As mentioned, it is known how to construct a PRF from any OWF, which is a minimal assumption [1], but this proof is much more involved.

Next, we will tackle a different problem: Suppose Bob has received a message claimed to be sent by Alice, but he is worried that perhaps an adversary sent the message and claims it is from Alice. Even if the message was encrypted using a CPA-secure encryption scheme with a

secret key that only Alice and Bob hold, this does not guarantee that the message was actually sent by Alice.

Until now, we were only concerned with the secrecy of their communication, the guarantee that nobody else can read Alice and Bob's messages. But we have not been concerned at all with the authenticity of their communication, the guarantee that nobody can forge a message in a way that makes it look like it was sent by Alice. This will be our focus next.

One solution that may come to mind, is to have Alice "sign" her message, i.e., add a (ridiculous) scribble to the end of her message. This of course does not work! An attacker can take a message "Attack at dawn" with Alice's scribble, and change it by only changing the word "dawn" to "dusk."

Not to mention that it is quite easy to copy these scribbles! It is surprising that it works reasonably well in practice.

A much much better solution would be for Alice to produce a signature that somehow depends on the message she is signing. In other words, for different messages, Alice should produce different signatures that the receiver is convinced that indeed it was Alice who sent the message, and the message was not altered by an adversary.

*Message Authentication Codes (*MAC*s)*

Definition 4. A MAC corresponding to a key space $\mathcal{K} = \{\mathcal{K}_\lambda\}_{\lambda\in\mathbb{N}}$[2] and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ is a poly-time computable function

$$\mathsf{MAC} : \mathcal{K}_\lambda \times \mathcal{M}_\lambda \to \{0,1\}^*$$

that satisfies the following security guarantee, referred to as *existential unforgeability against adaptive chosen message attacks*:

[2] We typically assume that $\mathcal{K}_\lambda = \{0,1\}^\lambda$.

Security:[3] For every poly-size adversary $\mathcal{A}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$ the adversary wins in the following game with probability $\leq \mu(\lambda)$:

[3] This security notion should sound familiar :-)

1. The challenger chooses a random key $k \leftarrow \mathcal{K}_\lambda$.
2. The adversary sends the challenger a message $m_i \in \mathcal{M}_\lambda$ and receives a "tag" $\tau_i = \mathsf{MAC}(k, m_i)$.

   This step can be repeated polynomially many times.
3. The adversary outputs (m^*, τ^*)
4. The adversary wins if $m^* \notin \{m_i\}$, where $\{m_i\}$ is the set of messages the adversary requested tags for, and if $\tau^* = \mathsf{MAC}(k, m^*)$.

Remark. A more concise way of writing the above security definition is as follows: For every poly-size $\mathcal{A}$ there exists a negligible μ such that for every $\lambda \in \mathbb{N}$

$$\Pr[\mathcal{A}^{\mathsf{MAC}(k,\cdot)}(1^\lambda) = (m^*, \tau^*) : \ \tau^* = \mathsf{MAC}(k, m^*) \ \ \wedge \ \ m^* \notin Q] \leq \mu(\lambda),$$

where Q denotes all the oracle queries that $\mathcal{A}$ sends its oracle.

Remark. There are many variants of this security definition. For example, a weaker security guarantee is *random* unforgeability against adaptive chosen message attacks, where the adversary wins if it generates a valid tag $\tau = \mathsf{MAC}(k, r)$ for a *random* message $r \leftarrow \mathcal{M}_\lambda$ chosen by the challenger, after seeing polynomially many tags for messages of its choice (chosen before r was given). Another, even weaker, security notion is random unforgeability against random chosen message attacks, where the adversary wins if it generates a valid tag $\tau = \mathsf{MAC}(k, r)$ for a random message $r \leftarrow \mathcal{M}_\lambda$ chosen by the challenger, after seeing tags for polynomially many *random* messages $r_1, \ldots, r_t \leftarrow \mathcal{M}_\lambda$, where $r^* \notin \{r_i\}_{i \in [t]}$.

Remark. Often a MAC is associated with a verification algorithm. With our definition, a tag τ on a message m is verified by checking that $\tau = \mathsf{MAC}(k, m)$. Separating the task of verifying a tag from the task of generating one, allows the task of verification to be possibly more efficient than the task of computing a MAC. In practice (and in our theoretical constructions) verification is done by recomputing the MAC.

Also, as with an encryption scheme, a MAC is often associated with a key generation algorithm. We assume for simplicity, without loss of generality, that the key is uniformly sampled from the key spaces.

MAC *Construction from any* PRF

The construction is extremely simple: Take any $\mathsf{PRF} : \mathcal{K}_\lambda \times \mathcal{M}_\lambda \to \{0,1\}^\lambda$. Define:

$$\mathsf{MAC}(k, m) = \mathsf{PRF}(k, m).$$

Theorem 5. *The above* MAC *is existentially unforgeable against adaptive chosen message attacks.*

Proof. Fix any poly-size $\mathcal{A}$. By the security of the PRF it holds that there exists a negligible μ such that for every $\lambda \in \mathbb{N}$,

$$\Pr[\mathcal{A}^{\mathsf{MAC}(k,\cdot)}(1^\lambda) = (m^*, \tau^*) : \ \tau^* = \mathsf{MAC}(k, m^*) \ \ \wedge \ \ m^* \notin Q] \leq$$
$$\Pr[\mathcal{A}^{R_\lambda(\cdot)}(1^\lambda) = (m^*, \tau^*) : \ \tau^* = \mathsf{MAC}(k, m^*) \ \ \wedge \ \ m^* \notin Q] + \mu(\lambda),$$

where $R_\lambda : \mathcal{M}_\lambda \to \{0,1\}^\lambda$ is a truly random function. It is easy to see that

$$\Pr[\mathcal{A}^{R_\lambda(\cdot)}(1^\lambda) = (m^*, \tau^*) : \ \tau^* = \mathsf{MAC}(k, m^*) \ \ \wedge \ \ m^* \notin Q] = 2^{-\lambda}.$$

Thus, we conclude that for every $\lambda \in \mathbb{N}$,

$$\Pr[\mathcal{A}^{\mathsf{MAC}(k,\cdot)}(1^\lambda) = (m^*, \tau^*) : \ \tau^* = \mathsf{MAC}(k, m^*) \ \ \wedge \ \ m^* \notin Q] \leq \mu(\lambda) + 2^{-\lambda}$$

which is negligible.

□

References

[1] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.