Random Oracles and Digital Signatures

Michele Ciampi

Introduction to Modern Cryptography, Lecture 16

Random Oracles

- A *random oracle* is a function that produces a random looking output for each query it receives.
- It must be consistent: if a question is repeated, the random oracle must return the same answer.
- Useful when abstracting a hash function in cryptographic applications.
- If a scheme is secure assuming the adversary views some hash function as a random oracle, it is said to be secure in the **Random Oracle Model**.

Random Oracles

- Given query M s.t. $(M, \cdot) \notin$ History, choose $t \xleftarrow{\$} Y$ and add (M, t) to History. Return t.
- Given query M s.t. $(M, t) \in$ History for some t, return t.

Figure: Hash function $H : \{0,1\}^* \longrightarrow Y$ modelled as a random oracle.

Random Oracles

- A scheme is designed and proven secure in the random-oracle model.
- In the real world, a random oracle is not available. Instead, the RO is instantiated with a hash function $\hat{H}$

Random Oracles

- If x has not been queried to H, then the value of $H(x)$ is uniform.
- If $\mathcal{A}$ queries x to H, the reduction can see this query and learn x. (Observability.)
- The reduction can set the value of $H(x)$ (i.e., the response to query x) to a value of its choice, as long as this value is correctly distributed, i.e., uniform. (Programmability.)

Objections to the RO model

- $\hat{H}$ cannot possibly be random (or even pseudorandom) since the adversary learns the description of $\hat{H}$. Hence, the value of the function on all inputs is immediately determined.
- Given that the description of $\hat{H}$ is given to the adversary, the adversary can query $\hat{H}$ locally. How can a reduction see the queries that the adversary makes, or program it?
- We do not have a clear idea of what it means for a concrete hash function to be "sufficiently good".

Support for the RO model

Why using the RO at all given all these problems?

- Efficient schemes
- A proof of security in the random-oracle model is significantly better than no proof at all.
- A proof of security for a scheme in the random-oracle model indicates that the scheme's design is "sound". If there is a problem is only because the hash fuction is not good enough.
- There have been no successful real-world attacks on schemes proven secure in the random-oracle model.

Digital signatures

- Digital signatures are technologically equivalent to hand-written signatures.
- A *signer* S has a unique private signing key and publishes the corresponding public verification key.
- S signs a message M and everyone who knows the public key can verify that M originated from the signer S.

Syntax

A **digital signature scheme** is a triple of algorithms as follows:

- The *key generation* algorithm $\text{Gen}(1^n)$ that outputs a signing (private) key sk and a verification (public) key vk.
- The *signing* algorithm $\text{Sign}(sk, M)$ that outputs a signature σ on message M.
- The *verification* algorithm $\text{Verify}(vk, M, \sigma)$ that outputs 1 if σ is valid and 0, otherwise.

Properties

- **Correctness:** For any message M in message space $\mathcal{M}$, it holds that

$$\Pr_{(sk,vk)\leftarrow \mathsf{Gen}(1^n)}\left[\mathsf{Verify}(vk, M, \mathsf{Sign}(sk, M)) = 1\right] \geq 1 - \mathsf{negl}(n) .$$

- **Unforgeability:** There exists no PPT adversary that can produce a valid message- signature pair without receiving it from external sources.

A formal definition of unforgeability

- Gen(1^n) is run to obtain keys (vk, sk).
- The adversary $\mathcal{A}$ is given vk and access to an oracle Sign$(sk, \cdot)$. The adversary outputs a pair (M, σ). Let $\mathcal{Q}$ denote the set of queries that $\mathcal{A}$ asked the oracle.
- $\mathcal{A}$ succeeds iff Verify$(vk, M, \sigma) = 1$ and $M \notin \mathcal{Q}$. In this case, output 1. Else, output 0.

Figure: The game $\mathrm{Game}_{\mathrm{EUF-CMA}}^{\mathcal{A}^{\mathsf{Sign}}}$.

We say that the digital signature scheme (Gen, Sign, Verify) has *existential unforgeability under adaptive chosen message attacks (EUF-CMA)* if for every PPT adversary $\mathcal{A}$, it holds that

$$\Pr\left[\mathrm{Game}_{\mathrm{EUF-CMA}}^{\mathcal{A}^{\mathsf{Sign}}}(1^n) = 1\right] \leq \mathsf{negl}(n) .$$

Trapdoor One-Way Functions

A *trapdoor one-way function (TOWF)* $f_e : X_e \longrightarrow Y_e$ with parameters $(e, z) \leftarrow \mathsf{Gen}_{\mathsf{TOWF}}(1^n)$ is a function that satisfies the following:

- *Easy to compute:* there exists a PPT algorithm that on input x returns $f_e(x)$.
- *Hard to invert:* for every PPT adversary $\mathcal{A}$

$$\Pr\left[x \stackrel{\$}{\leftarrow} X_e; \mathcal{A}(e, f_e(x)) \in f_e^{-1}(f_e(x))\right] \leq \mathsf{negl}(n) \ .$$

- *Easy to invert with trapdoor:* There exists PPT algorithm $\mathcal{T}$ such that

$$\mathcal{T}(e, z, f_e(x)) \in f_e^{-1}(f_e(x)) \ .$$

Digital signatures from trapdoor one-way functions

Let $H : \{0,1\}^* \longrightarrow Y_e$ be a (collision resistant) hash function and $f_e : X_e \longrightarrow Y_e$ be a TOWF with parameter generation algorithm G_{TOWF} and trapdoor algorithm $\mathcal{T}$. We define the following signature scheme:

- $\mathsf{Gen}(1^n)$: $(e, z) \leftarrow \mathsf{Gen}_{\mathsf{TOWF}}(1^n)$. Output $vk := e$ and $sk := (e, z)$.
- $\mathsf{Sign}(sk, M)$: $h \leftarrow H(M)$; $\sigma \leftarrow \mathcal{T}(e, z, h)$.
- $\mathsf{Verify}(vk, M, \sigma)$: If $f_e(\sigma) = H(M)$ output 1. Else, output 0.

Figure: Digital signatures from trapdoor one-way functions.

Correctness

For any message M, we have that $h \leftarrow H(M)$ and $\sigma \leftarrow \mathcal{T}(e, z, h)$, so $\sigma \in f_e^{-1}(h) = f_e^{-1}(H(M))$. Therefore,

$$f_e(\sigma) = H(M) \; .$$

Unforgeability

Theorem

Suppose that $f_e : X_e \longrightarrow Y_e$ *is bijective and* $H : \{0,1\}^* \longrightarrow Y_e$ *is a random oracle. Suppose that* $|Y_e| \geq 2^n$*. Then for every PPT adversary* $\mathcal{A}$ *that breaks the EUF-CMA security of* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *with probability* α*, i.e.,*

$$\Pr\left[\mathrm{Game}_{\mathrm{EUF-CMA}}^{\mathcal{A}^{\mathsf{Sign}}}(1^n) = 1\right] = \alpha \ ,$$

there exists a PPT adversary $\mathcal{B}$ *that breaks the one-way property of* f_e*, i.e.,*

$$\Pr\left[x \xleftarrow{\$} X_e; \mathcal{B}(e, f_e(x)) = x\right] \geq \frac{1}{q_H}\left(\alpha - \frac{1}{2^n}\right) \ ,$$

where q_H *is the number of queries* $\mathcal{A}$ *makes to the random oracle* H*.*

Proof of EUF-CMA security

- Let $(e, z) \leftarrow \mathsf{Gen}_{\mathsf{TOWF}}(1^n)$, $x \xleftarrow{\$} X_e$ and $y = f_e(x)$. Since f_e is a bijection, $\mathcal{B}$ is given (e, y) and its goal is to find $x = f_e^{-1}(y)$.
- The adversary $\mathcal{B}$ must simulate the oracles H and Sign to use adversary $\mathcal{A}$.

Proof of EUF-CMA security

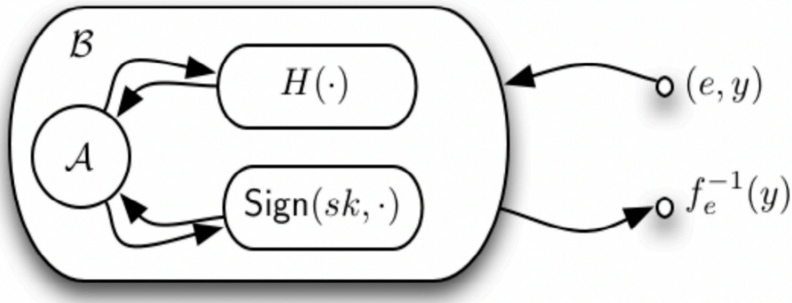

Figure: The adversary $\mathcal{B}$ must simulate H and Sign to use adversary A.

Proof of EUF-CMA security

- First, suppose that $\mathcal{A}$ makes no signing queries, so it produces (M^*, σ^*) after making q_H queries to the random oracle.
- $\mathcal{B}$ will simulate the random oracle by plugging in y into the oracle's responses.

Choose $j \xleftarrow{\$} \{1, 2, \ldots, q_H\}$.

- Given query M s.t. $(M, \cdot) \notin$ History: if this is the jth query, set $t = y$, else choose $t \xleftarrow{\$} Y_e$. Add (M, t) to History. Return t.
- Given query M s.t. $(M, t) \in$ History for some t, return t.

Figure: Modified random oracle simulation by $\mathcal{B}$.

Proof of EUF-CMA security

Let E be the event that $(M^*, \cdot) \in \text{History}$, i.e. $\mathcal{A}$ asks M^* to H. Then,

$$\Pr\left[\mathcal{A} \text{ succeeds } \middle| \neg E\right] \leq \frac{1}{|Y_e|} \leq \frac{1}{2^n} \; .$$

This is the case since given the event $\neg E$, the adversary has not asked M^* to H and thus the value of $H(M^*)$ is undetermined until the final step of $\mathcal{B}$ takes place. Thus,
$\Pr\left[f_e(\sigma^*) = H(M^*) \middle| \neg E\right] = \frac{1}{|Y_e|} \leq \frac{1}{2^n}$.
Consequently,

$$\begin{aligned}
\Pr\left[\mathcal{A} \text{ succeeds} \wedge E\right] &\overset{LoTP}{=} \Pr\left[\mathcal{A} \text{ succeeds}\right] - \Pr\left[\mathcal{A} \text{ succeeds} \wedge \neg E\right] \overset{CP}{=} \\
&= \Pr\left[\mathcal{A} \text{ succeeds}\right] - \Pr\left[\mathcal{A} \text{ succeeds } \middle| \neg E\right] \Pr\left[\neg E\right] \\
&\geq \Pr\left[\mathcal{A} \text{ succeeds}\right] - \Pr\left[\mathcal{A} \text{ succeeds } \middle| \neg E\right] \geq \\
&\geq \alpha - \frac{1}{2^n} \; .
\end{aligned}$$

Conditional Independency

The events A and B are said to be conditionally independent with respect to the event C, if and only if

$$\Pr\left[A|B \wedge C\right] = \Pr\left[A|C\right]$$

Equivalent form:

$$\Pr\left[A \wedge B|C\right] = \Pr\left[A|C\right]\Pr\left[B|C\right]$$

Proof of EUF-CMA security

Given event E, let G be the event that the random oracle simulation will guess correctly the query that M^* is asked. We have that $\Pr[G|E] = \frac{1}{q_H}$.

Proof of EUF-CMA security

Given event E, let G be the event that the random oracle simulation will guess correctly the query that M^* is asked. We have that $\Pr[G|E] = \frac{1}{q_H}$.

If G occurs, then $H(M^*) = y$. If additionally $\mathcal{A}$ succeeds, then $f_e(\sigma^*) = H(M^*) = y$, i.e., σ^* is a preimage of y! So, $\mathcal{B}$ succeeds by returning $\sigma^* = x$.

Due to the independence of G and the success of $\mathcal{A}$ in the conditional space E, we have that

$$\begin{aligned}
\Pr\left[\mathcal{B}\text{ succeeds}\right] &\overset{LTP}{=} \Pr\left[\mathcal{B}\text{ succeeds} \wedge E\right] + \Pr\left[\mathcal{B}\text{ succeeds} \wedge \neg E\right] \overset{Cond.P.}{=} \\
&= \Pr\left[\mathcal{B}\text{ succeeds}\middle| E\right] \cdot \Pr[E] + \\
&\qquad \Pr\left[\mathcal{B}\text{ succeeds}\middle| \neg E\right] \cdot \Pr[\neg E] \geq \\
&\geq \Pr\left[\mathcal{B}\text{ succeeds}\middle| E\right] \cdot \Pr[E] \geq \\
&\geq \Pr\left[\mathcal{A}\text{ succeeds} \wedge G\middle| E\right] \cdot \Pr[E] \overset{Cond.Ind.}{=} \\
&= \Pr\left[\mathcal{A}\text{ succeeds}\middle| E\right] \cdot \Pr[G|E] \cdot \Pr[E] \overset{Cond.P.}{=} \\
&= \Pr\left[\mathcal{A}\text{ succeeds} \wedge E\right] \cdot \Pr[G|E] \geq \frac{1}{q_H}\left(\alpha - \frac{1}{2^n}\right).
\end{aligned}$$

Proof of EUF-CMA security

Consider the general case where $\mathcal{A}$ makes (polynomially many) queries to the signing oracle. $\mathcal{B}$ must answer in a way that is consistent with the random oracle queries.

Choose $j \xleftarrow{\$} \{1, 2, \ldots, q_H\}$.

- Given query M s.t. $(M, \cdot, \cdot) \notin$ History: if this is the jth query, set $t = y$, $\rho = \perp$. Else, choose $\rho \xleftarrow{\$} X_e$ and set $t = f_e(\rho)$. Add (M, t, ρ) to History. Return t.
- Given query M s.t. $(M, t, \rho) \in$ History for some t, return t.

Figure: A second modified random oracle simulation as used by algorithm $\mathcal{B}$ to "plug-in" a challenge y into the oracle's responses while keeping the "pre-images" of the oracles responses under the map f_e.

Proof of EUF-CMA security

- When asked to sign M, $\mathcal{B}$ can first ask its random oracle for M and look for (M, t, ρ) in History and, unless $\rho = \perp$, proceed to answer the query with ρ. By construction, $f_e(\rho) = t = H(M)$, so ρ is valid.
- The case $\rho = \perp$ means that the guess of $\mathcal{B}$ for j is mistaken (due to the condition that a successful forgery must be on a message that $\mathcal{A}$ does not query to the signing oracle) and thus the simulation of $\mathcal{B}$ will fail. We call this event F.
- It holds that $(\mathcal{A}$ succeeds$) \cap G \cap F = \emptyset$.

Proof of EUF-CMA security

As previously, we have that

$$\Pr\left[\mathcal{A} \text{ succeeds} \wedge E\right] \geq \alpha - \frac{1}{2^n}$$

In addition, since $(\mathcal{A} \text{ succeeds}) \cap G \cap F = \emptyset$, it holds that

$$\Pr\left[\mathcal{A} \text{ succeeds} \wedge G \wedge E \wedge \neg F\right] = \Pr\left[\mathcal{A} \text{ succeeds} \wedge G \wedge E\right] .$$

Proof of EUF-CMA security

Therefore, we get that

$$
\begin{aligned}
\Pr\left[\mathcal{B} \text{ succeeds}\right] &\geq \Pr\left[\mathcal{A} \text{ succeeds} \wedge G \wedge E \wedge \neg F\right] = \\
&= \Pr\left[\mathcal{A} \text{ succeeds} \wedge G \wedge E\right] \overset{Cond.P.}{=} \\
&= \Pr\left[\mathcal{A} \text{ succeeds} \wedge G \middle| E\right] \cdot \Pr[E] = \\
&= \Pr\left[\mathcal{A} \text{ succeeds} \middle| E\right] \cdot \Pr[G|E] \cdot \Pr[E] = \\
&= \Pr\left[\mathcal{A} \text{ succeeds} \wedge E\right] \cdot \Pr[G|E] \geq \\
&\geq \frac{1}{q_H}\left(\alpha - \frac{1}{2^n}\right) .
\end{aligned}
$$

Proof of EUF-CMA security

The modified random oracle that $\mathcal{B}$ manages is indistinguishable from an original random oracle.

- Since $f_e(\cdot)$ is a bijection, $f_e(\rho) = t$ is uniformly distributed over Y_e when ρ is uniformly distributed over X_e.
- As for the jth query, recall that the input y of $\mathcal{B}$ is uniformly distributed over Y_e (since $y = f_e(x)$ and $x \stackrel{\$}{\leftarrow} X_e$).

$\square$

Instantiation: RSA full-domain hash signatures

- Gen: On input 1^n choose two n-bit random primes p and q. Compute $N = pq$ and $\phi(N) = (p-1)(q-1)$. Choose $e > 1$ such that $gcd(e, \phi(N)) = 1$. Compute $d := e^{-1} \bmod \phi(N)$. Return (N, e) as the verification key and (N, d) as the signing key. A full-domain hash function H is available to all parties.
- Sign: on input a signing key (N, d) and a message M, output the digital signature

$$\sigma = H(M)^d \bmod N \ .$$

- Verify: on input a verification key (N, e) and (M, σ), verify that $\sigma^e = H(M) \bmod N$. If equality holds, the result is `True`; otherwise, the result is `False`.

Figure: RSA-FDH signatures.

End

References: -From Introduction to Modern Cryptography: Sec. 5.5 (this is a discussion on the random oracle model). -From Prof. Kiayias's lecture notes: Section 7 (pages 42-46), Section 7 (pages 45-47).