# Cryptography

Lecture 12

# Announcements

- HW5 due 3/13
- Midterm Upcoming on 3/15
  - Review sheet will be posted on course webpage by tonight
  - Solutions and Cheat Sheet posted soon on Canvas

# Agenda

- Last time:
  - Domain Extension for MACs (K/L 4.4) and Class Exercise solutions
  - CCA security (K/L 3.7)
  - Unforgeability for Encryption (K/L 4.5)
- This time:
  - Authenticated Encryption (K/L 4.5)
  - Collision-Resistant Hash Functions (K/L 5.1)
  - Hash-and-Mac
  - Domain extension for CRHF

# Chosen Ciphertext Security

# CCA Security

Consider a private-key encryption scheme $\Pi = (Gen, Enc, Dec)$, any adversary $A$, and any value $n$ for the security parameter.

Experiment $PrivK_{A,\Pi}^{cca}(n)$

Adversary $A(1^n)$ | Challenger

$k \leftarrow Gen(1^n)$

$A^{Enc_k(\cdot),Dec_k(\cdot)}$ $\xrightarrow{m'/c'}$

$\xleftarrow{c'/m'}$

$\vdots$

$\xrightarrow{m_0, m_1}$

$b \leftarrow \{0,1\}$

$c \leftarrow Enc_k(m_b)$

$\xleftarrow{c}$

$A^{Enc_k(\cdot),Dec_k^*(\cdot)}$ $\xrightarrow{m'/c'}$

$\xleftarrow{c'/m'}$

$\vdots$

$\xrightarrow{b'}$

$A$ may not query challenge ciphertext $c$ to the decryption oracle

$PrivK_{A,\Pi}^{cca}(n) = 1$ if $b' = b$ and $PrivK_{A,\Pi}^{cca}(n) = 0$ if $b' \neq b$.

# CCA Security

The CCA Indistinguishability Experiment $PrivK^{cca}{}_{A,\Pi}(n)$:

1. A key $k$ is generated by running $Gen(1^n)$.
2. The adversary $A$ is given input $1^n$ and oracle access to $Enc_k(\cdot)$ and $Dec_k(\cdot)$, and outputs a pair of messages $m_0, m_1$ of the same length.
3. A random bit $b \leftarrow \{0,1\}$ is chosen, and then a challenge ciphertext $c \leftarrow Enc_k(m_b)$ is computed and given to $A$.
4. The adversary $A$ continues to have oracle access to $Enc_k(\cdot)$ and $Dec_k(\cdot)$, but is not allowed to query the latter on the challenge ciphertext itself. Eventually, $A$ outputs a bit $b'$.
5. The output of the experiment is defined to be $1$ if $b' = b$, and $0$ otherwise.

# CCA Security

A private-key encryption scheme $\Pi = (Gen, Enc, Dec)$ has indistinguishable encryptions under a chosen-ciphertext attack if for all ppt adversaries $A$ there exists a negligible function $negl$ such that

$$\Pr\left[PrivK^{cca}{}_{A,\Pi}(n) = 1\right] \leq \frac{1}{2} + negl(n),$$

where the probability is taken over the random coins used by $A$, as well as the random coins used in the experiment.

# Authenticated Encryption

The unforgeable encryption experiment $EncForge_{A,\Pi}(n)$: (similar to MAC game)

1. Run $Gen(1^n)$ to obtain key $k$.
2. The adversary $A$ is given input $1^n$ and access to an encryption oracle $Enc_k(\cdot)$ (MAC). The adversary outputs a ciphertext $c$. (similar to forging)
3. Let $m := Dec_k(c)$, and let $Q$ denote the set of all queries that $A$ asked its encryption oracle. The output of the experiment is 1 if and only if (1) $m \neq \perp$ and (2) $m \notin Q$.

"\bot"
"\perp"

for Auth Enc, when dec we might get "$\perp$".

# $\boxed{\text{Authenticated Encryption}}$

Definition: A private-key encryption scheme $\Pi$ is unforgeable if for all ppt adversaries $A$, there is a negligible funcion $neg$ such that:

$$\Pr\left[EncForge_{A,\Pi}(n) = 1\right] \leq neg(n).$$

Definition: A private-key encryption scheme is an authenticated encryption scheme if it is CCA-secure and unforgeable.

Intuition: We want to combine Enc and Mac in a secure way. Resulting scheme, both privacy + (integrity)

# Generic Constructions

Cauth

NOT SECURE

# Encrypt-and-authenticate

Always choose independent keys for Enc, and Mac

Encryption and message authentication are computed independently in parallel.

$$c \leftarrow Enc_{k_E}(m) \qquad t \leftarrow Mac_{k_M}(m)$$

$$\langle c, t \rangle$$

leak inf. about messge

Is this secure?

Not in theory

what about in practice?

Any det. MAC → NOT CPA secure

# Encrypt-and-authenticate

Encryption and message authentication are computed independently in parallel.

$$c \leftarrow Enc_{k_E}(m) \qquad t \leftarrow Mac_{k_M}(m)$$

$$\langle c, t \rangle$$

Is this secure? NO! Tag can leak info on $m$

NOT SECURE

# Authenticate-then-encrypt

Here a MAC tag $t$ is first computed, and then the message and tag are encrypted together.

$$t \leftarrow \underbrace{Mac_{k_M}}_{\text{Exis. Unfor in the presence of CMA attack.}}(m) \qquad c \leftarrow \underbrace{Enc_{k_E}}_{\text{Any CPA-secure}}(m||t)$$

$c$ is sent

Is this secure?

Not nec. CCA secure

"malleable"

Doesn't achieve our notion of Auth Enc

# Authenticate-then-encrypt

Here a MAC tag $t$ is first computed, and then the message and tag are encrypted together.

$$t \leftarrow Mac_{k_M}(m) \qquad c \leftarrow Enc_{k_E}(m||t)$$

$c$ is sent

Is this secure? NO! Encryption scheme may not be CCA-secure.

# Encrypt-then-authenticate

The message $m$ is first encrypted and then a MAC tag is computed over the result

$$c \leftarrow Enc_{k_E}(m) \quad t \leftarrow Mac_{k_M}(c)$$

$$\langle c, t \rangle$$

Is this secure? Yes, & assuming $k_E, k_M$ are generated independently

# Encrypt-then-authenticate

The message $m$ is first encrypted and then a MAC tag is computed over the result

$$c \leftarrow Enc_{k_E}(m) \quad t \leftarrow Mac_{k_M}(c)$$

$$\langle c, t \rangle$$

Is this secure? YES! As long as the MAC is strongly secure.

In order to get CCA security

# Collision Resistant Hashing

Bitcoin.

Cryptographic hash functions

SHA-2 < SHA-256

$H(\underbrace{m}_{\text{any length}}) = \boxed{y}$ output is alway 256 bits.

Collision Resistance

Security guarantee: Adv cannot find 2 messages $m, m'$ s.t. $H(m) = H(m')$

# Collision Resistant Hashing

Definition: A hash function (with output length $\ell$) is a pair of ppt algorithms $(Gen, H)$ satisfying the following:

- $\boxed{Gen}$ takes as input a security parameter $1^n$ and outputs a key $s$. We assume that $1^n$ is implicit in s.

  key is public
- $H$ takes as input a key $s$ and a string $x \in \{0,1\}^*$ and outputs a string $H^s(x) \in \{0,1\}^{\ell(n)}$.

256

If $H^s$ is defined only for inputs $x \in \{0,1\}^{\ell'(n)}$ and
512 $\boxed{\ell'(n)} > \boxed{\ell(n)}$, then we say that $(Gen, H)$ is a fixed-length hash function for inputs of length $\ell'$. In this case, we also call $H$ a compression function.

# The collision-finding experiment

$Hashcoll_{A,\Pi}(n)$:

1. A key $s$ is generated by running $Gen(1^n)$.
2. The adversary $A$ is given $\boxed{s}$ and outputs $x, x'$. (If $\Pi$ is a fixed-length hash function for inputs of length $\ell'(n)$, then we require $x, x' \in \{0,1\}^{\ell'(n)}$.)
3. The output of the experiment is defined to be 1 if and only if $x \neq x'$ and $H^s(x) = H^s(x')$. In such a case we say that $A$ has found a collision.

Given $H(x) = y$ find some pre-image $x'$ s.t. $H(x') = y$.

# Security Definition

Definition: A hash function $\Pi = (Gen, H)$ is collision resistant if for all ppt adversaries $A$ there is a negligible function $neg$ such that

$$\Pr\left[Hashcoll_{A,\Pi}(n) = 1\right] \leq neg(n).$$

# Message Authentication Using Hash Functions

Recap: Mac fixed-length messages.

$m \in \{0,1\}^n$

$$\mathrm{Mac}_k(m) = F_k(m)$$

Hash-and-Mac: "Domain extension" for Macs.

lift a Mac for fixed-length msgs $\longrightarrow$ Mac for arbit length

$$F_k(\underbrace{H(m)}_{n}) = t.$$

# Hash-and-Mac Construction

Let $\Pi = \boxed{(Mac, Vrfy)}$ be a MAC for messages of length $\boxed{\ell(n)}$, and let $\Pi_H = (Gen_H, H)$ be a hash function with output length $\ell(n)$. Construct a MAC $\Pi' = (Gen', Mac', Vrfy')$ for arbitrary-length messages as follows:

- $Gen'$: on input $1^n$, choose uniform $k \in \{0,1\}^n$ and run $Gen_H(1^n)$ to obtain $s$. The key is $k' := \langle k, s \rangle$.
- $Mac'$: on input a key $\langle k, s \rangle$ and a message $m \in \{0,1\}^*$, output $\boxed{t \leftarrow Mac_k(H^s(m)).}$
- $Vrfy'$: on input a key $\langle k, s \rangle$, a message $m \in \{0,1\}^*$, and a MAC tag $t$, output 1 if and only if $Vrfy_k(H^s(m), t) = 1$.

# Security of Hash-and-MAC

Theorem: If $\Pi$ is a secure MAC for messages of length $\ell$ and $\Pi_H$ is collision resistant, then the construction above is a secure MAC for arbitrary-length messages.

# Proof Intuition

Let $Q$ be the set of messages $m$ queried by adversary $A$.

Assume $A$ manages to forge a tag for a message $m^* \notin Q$.

There are two cases to consider:

1. $H^s(m^*) = H^s(m)$ for some message $m \in Q$. Then $A$ breaks collision resistance of $H^s$.
2. $\boxed{H^s(m^*)} \neq H^s(m)$ for all messages $m \in Q$. Then $A$ forges a valid tag with respect to MAC $\Pi$.

underlying fixed length MAC.

$m^*$?

$\boxed{H(m^*)}$ ?

$$\boxed{\underline{t^*} = Mac_k(\underbrace{H(m^*)})}$$

$m_1$, $m_2$, $m_3$

$\underline{H(m_1)}$, $\underline{H(m_2)}$, $\underline{H(m_3)}$

Queries made to $Mac_K(H(\cdot))$

$(H(m^*), t^*)$ is a forgery on $Mac_K$.