

Cryptography

Lecture 11

Announcements

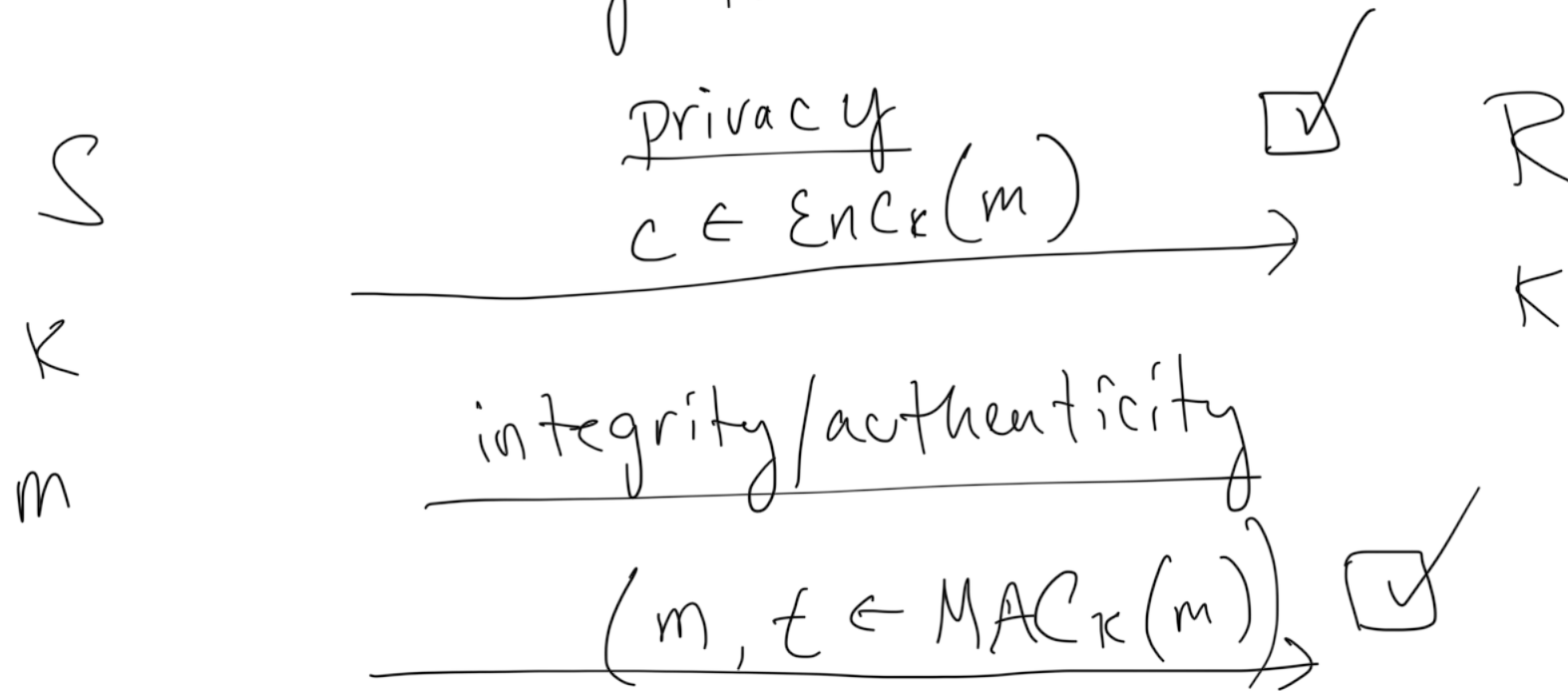
- HW3 due 3/6

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

Big Picture



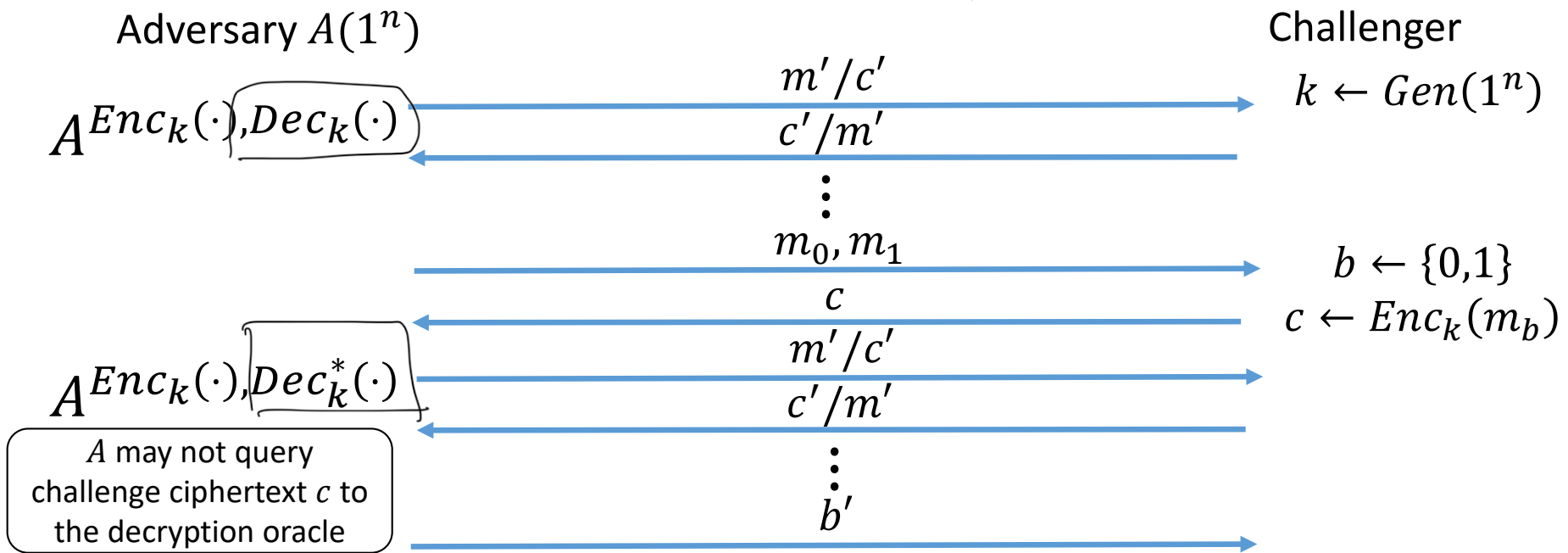
How do we properly combine Enc + MAC to
simultaneously achieve privacy, integrity + authenticity

Auth Enc

CCA Security

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

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



$PrivK_{A,\Pi}^{cca}(n) = 1$ 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.

Unforgeability

~~Authenticated Encryption~~

The unforgeable encryption experiment

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

MAC security 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 $Enc_k(\cdot)$. The adversary outputs a ciphertext (c) .
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) $m \notin Q$.

\perp bot

\perp perp

prevents a replay attack

~~Authenticated Encryption~~

Unforgeability

Definition: A private-key encryption scheme Π is unforgeable if for all ppt adversaries A , there is a negligible function neg such that:

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

Authenticated Encryption

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

Generic Constructions

& must use independently generated keys for Enc + MAC
in all these constructions

Encrypt-and-authenticate

Encryption and message authentication are
computed independently in parallel.

$$c \leftarrow \text{Enc}_{k_E}(m) \quad \textcircled{t} \leftarrow \text{Mac}_{k_M}(m)$$

$\boxed{\langle c, t \rangle}$

} Enc alg
for
Auth Enc
scheme

ciphertext Auth Enc

Is this secure?

↓ deterministic MAC


not CPA secure

$$m \parallel F_K(m)$$

Encrypt-and-authenticate

Encryption and message authentication are computed independently in parallel.

$$c \leftarrow \text{Enc}_{k_E}(m) \quad t \leftarrow \text{Mac}_{k_M}(m)$$

$\langle c, t \rangle$ 

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.

$$t \leftarrow \text{Mac}_{k_M}(m) \quad c \leftarrow \text{Enc}_{k_E}(m||t)$$

c is sent

} Auth Enc
Scheme

} Description
of Enc
Alg

Is this secure?

Authenticate-then-encrypt

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

$$t \leftarrow \text{Mac}_{k_M}(m) \quad c \leftarrow \text{Enc}_{k_E}(m||t)$$

c is sent

CCA secure.

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

$$c \leftarrow \text{Enc}_{k_E}(m) \quad t \leftarrow \text{Mac}_{k_M}(c)$$

$\boxed{\langle c, t \rangle}$

} Enc Alg
for
Auth Enc

Is this secure?

Encrypt-then-authenticate

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

$$c \leftarrow \text{Enc}_{k_E}(m) \quad t \leftarrow \text{Mac}_{k_M}(c)$$
$$\langle c, t \rangle$$

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

(c, t')

Cannot produce a different tag t for a ciphertext you've already seen.

Collision Resistant Hashing

H

actively try to find 2 inputs

x_1, x_2

$$H(x_1) = H(x_2)$$

Collision Resistant Hashing

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

- Gen takes as input a security parameter 1^n and outputs a key s . We assume that 1^n is implicit in s .
- H takes as input a key s and a string $x \in \{0,1\}^*$ and outputs a string $H^s(x) \in \{0,1\}^{\ell(n)}$.

$$H^s(\cdot)$$

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

The collision-finding experiment

*Hashcoll*_{A,Π}(*n*):

1. A key s is generated by running $Gen(1^n)$.
2. The adversary A is given s and outputs x, x' . (If Π 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.

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

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

$\underbrace{\hspace{1em}}$
choice of (s) ,
random coins of A

Message Authentication Using Hash Functions

Fixed length MAC $\Pi_{MAC} = (\text{Gen}, \text{MAC}, \text{Vrfy})$
 $\Pi_H: (\text{Gen}, \text{CRHF } H)$

Construction: m

$$t = \text{MAC}_k \left(\underbrace{H^s(m)}_{128} \right)$$

Alternative
to CBC-MAC

Hash-and-Mac Construction

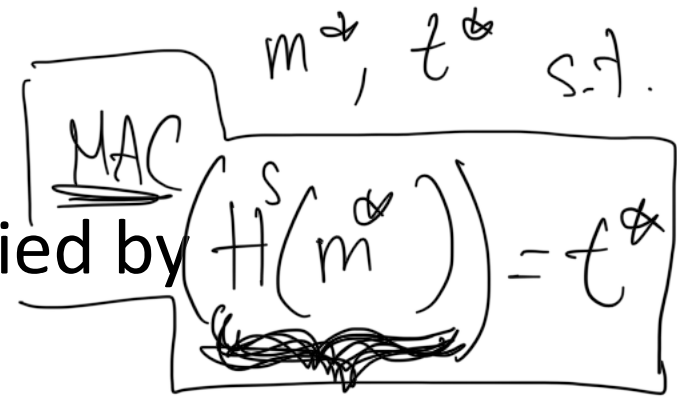
Let $\Pi = (Mac, Vrfy)$ be a MAC for messages of length $\ell(n)$, and let $\Pi_H = (Gen_H, H)$ be a hash function with output length $\ell(n)$. Construct a MAC $\Pi' = (Gen', Mac', Vrfy')$ for arbitrary-length messages as follows:

- 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 $t \leftarrow Mac_k(H^s(m))$.
- $Vrfy'$: 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 Π is a secure MAC for messages of length ℓ and Π_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. $H^s(m^*) \neq H^s(m)$ for all messages $m \in Q$.

Then A forges a valid tag with respect to MAC Π .

Domain Extension

The Merkle-Damgård Transform

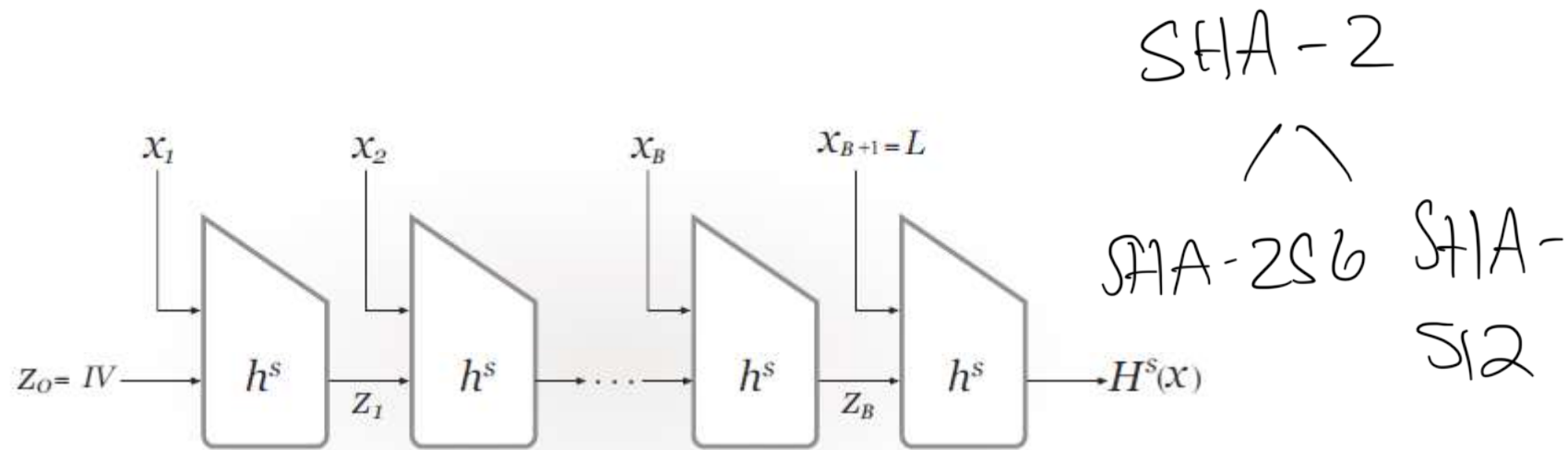


FIGURE 5.1: The Merkle-Damgård transform.

The Merkle-Damgard Transform

Let (Gen, h) be a fixed-length hash function for inputs of length $2n$ and with output length n . Construct hash function (Gen, H) as follows:

- Gen : remains unchanged
- H : on input a key s and a string $x \in \{0,1\}^*$ of length $L < 2^n$, do the following:
 1. Set $B := \left\lceil \frac{L}{n} \right\rceil$ (i.e., the number of blocks in x). Pad x with zeros so its length is a multiple of n . Parse the padded result as the sequence of n -bit blocks x_1, \dots, x_B . Set $x_{B+1} := L$, where L is encoded as an n -bit string.
 2. Set $z_0 := 0^n$. (This is also called the IV.)
 3. For $i = 1, \dots, B + 1$, compute $z_i := h^s(z_{i-1} || x_i)$.
 4. Output z_{B+1} .

Security of Merkle-Damgard

Theorem: If (Gen, h) is collision resistant, then so is (Gen, H) .