Cryptography

Lecture 5

Arpita Patra

Quick Recall and Today's Roadmap

- CCA Security, more stronger than CPA security
- Break of CBC Mode CPA secure scheme under CCA- Padding Oracle Attack
- MAC
- Security Definitions: CMA, sCMA. CMVA, sCMVA

- PRF-based MAC
- Domain Extension for MAC: To handle arbitrary length message
  - Not at all an easy task;
  - Naïve construction (by Goldreich);
  - Proof of Security
  - CBC-MAC: Practical Domain Extension
- Authenticated Encryption: Privacy and Integrity
  - Notion that subsumes CCA-security
  - Construction (again a bit tricky)
  - proof of Security

CMA Security for MAC

Experiment $\text{Mac-forge}^{\text{cma}}_{A,\Pi}(n)$

Π = (Gen, Mac, Vrfy), n

Attacker A

I can break Π

Run time: Poly(n)

$Q = \{(m_1, \ldots, m_l)\}$

Training Phase

(m, t)

Forged tag generated by A

Let me verify

k

Gen(1^n)

game output

- 1 (A succeeds) if $\text{Vrfy}_k(m, t) = 1$ and $m \notin Q$
- 0 (A fails) otherwise

Π is CMA- secure if for every A, there is a negl(n) such that

$$\Pr[\text{Mac-forge}^{\text{cma}}_{A,\Pi}(n) = 1] \leq \text{negl}(n)$$

Strong CMA Security for MAC

Experiment $\text{Mac-sforge}^{\text{cma}}_{A,\Pi}(n)$

Π = (Gen, Mac, Vrfy), n

Attacker A

I can break Π

Run time: Poly(n)

$Q = \{(m_1, t_1), \ldots, (m_l, t_l)\}$

Training Phase

(m, t)

Forged tag generated by A

Let me verify

k

$\text{Gen}(1^n)$

game output

- 1 (A succeeds) if $\text{Vrfy}_k(m, t) = 1$ and $(m, t) \notin Q$
- 0 (A fails) otherwise

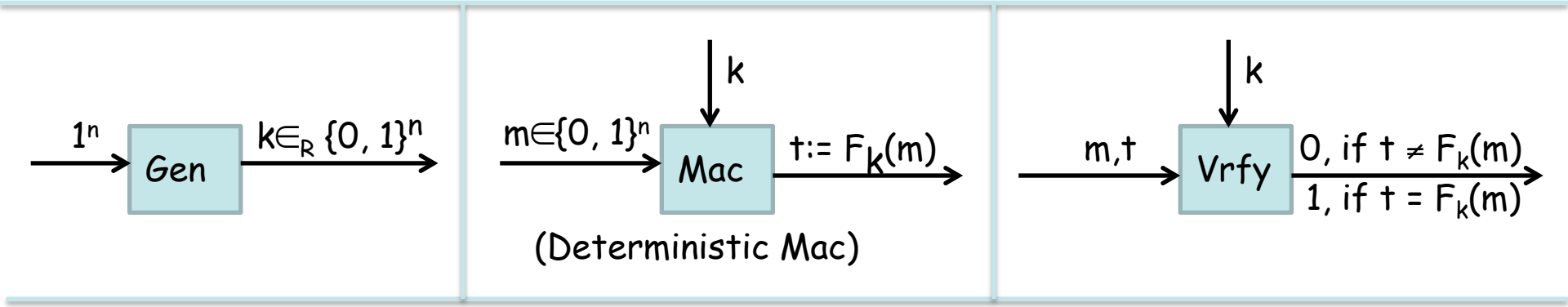
Π is strong CMA-secure if for every A, there is a negl(n) such that

$$\Pr[\text{Mac-sforge}^{\text{cma}}_{A,\Pi}(n) = 1] \leq \text{negl}(n)$$

Fixed-length MAC from PRF

- Let $F:\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a PRF

Then Π = (Gen, Mac, Vrfy) is a fixed-length MAC for n-bit strings where :



Theorem: If F is a PRF then Π is a CMA-secure MAC.

- Show that if Π is not CMA-secure then F is not a PRF by designing a distinguisher for F
- If instead a TRF f was used to compute tag then an attacker can guess f(m) for a "new" m with probability at most 2^{-n}
- The same should hold even if a PRF is used (as key is unknown)

Domain Extension

Given a scheme that handles fixed-length message.
How to handle arbitrary-length messages

SKE

Break the message into blocks and encrypt each block using fixed-length scheme (minimum security notion CPA-security)

Want efficiency?– Go for Mode of operations

MAC

The same does not work here– Additional tricks necessary

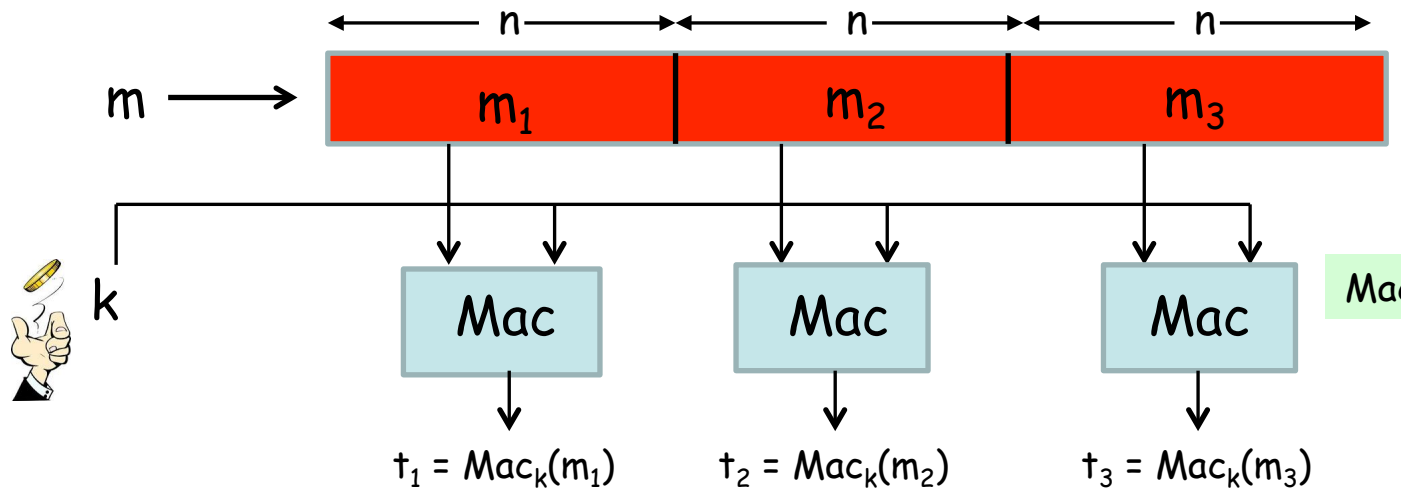
Want efficiency?– CBC-MAC, C-MAC, Hash-and-MAC, HMAC

Domain Extension

Warning!! Simple ideas do not work !!

Attempt I

- Divide the message into blocks and authenticate each separately via fixed-length MAC



$Mac_k(m) = t = t_1 \,||\, t_2 \,||\, t_3$

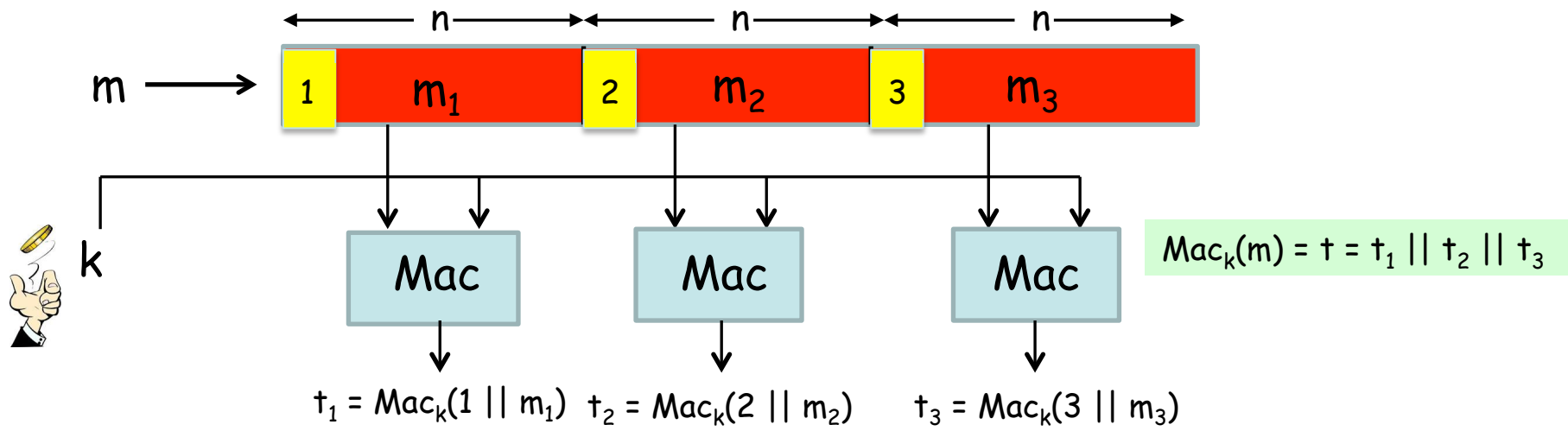
- Block re-ordering attack :
  - Given (m, t), where $m = m_1 \,||\, m_2 \,||\, m_3$ and $t = t_1 \,||\, t_2 \,||\, t_3$
  - Then (m', t') is a valid pair, where $m' = m_2 \,||\, m_1 \,||\, m_3$ and $t' = t_2 \,||\, t_1 \,||\, t_3$

Domain Extension for MAC

Warning!! Simple ideas do not work !!

Attempt II

- Prevent the previous attack by authenticating block index along with each block



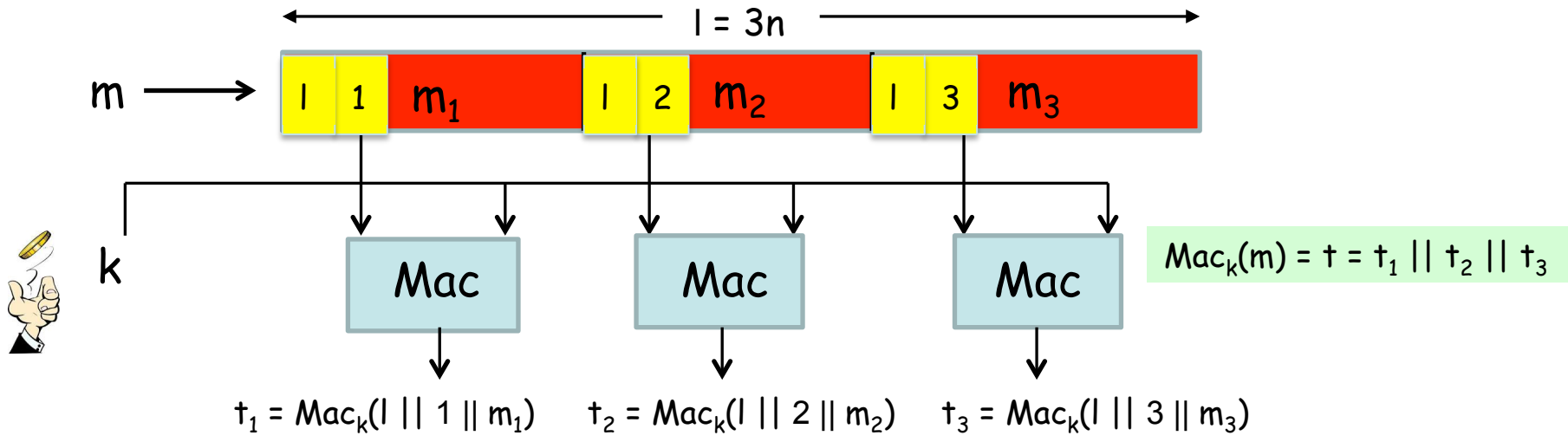
- Truncation attack :
  - A valid (msg, tag) pair can be generated by dropping (msg, tag) blocks from the end
  - $(m_1 || m_2, t_1 || t_2)$ is a valid new (msg, tag) pair generated from $(m_1 || m_2 || m_3, t_1 || t_2 || t_3)$

Domain Extension for MAC

Warning!! Simple ideas do not work !!

Attempt III

- Prevent the previous attack by additionally authenticating message length with each block



- Mix-and-match attack :
  - Suppose attacker learns $(m_1 \,||\, m_2 \,||\, m_3, t_1 \,||\, t_2 \,||\, t_3)$ and $(m'_1 \,||\, m'_2 \,||\, m'_3, t'_1 \,||\, t'_2 \,||\, t'_3)$ where $(m_1 \,||\, m_2 \,||\, m_3) = (m'_1 \,||\, m'_2 \,||\, m'_3)$
  - Then $(m_1 \,||\, m'_2 \,||\, m_3, t_1 \,||\, t'_2 \,||\, t_3)$ is a valid, new (message, tag) pair

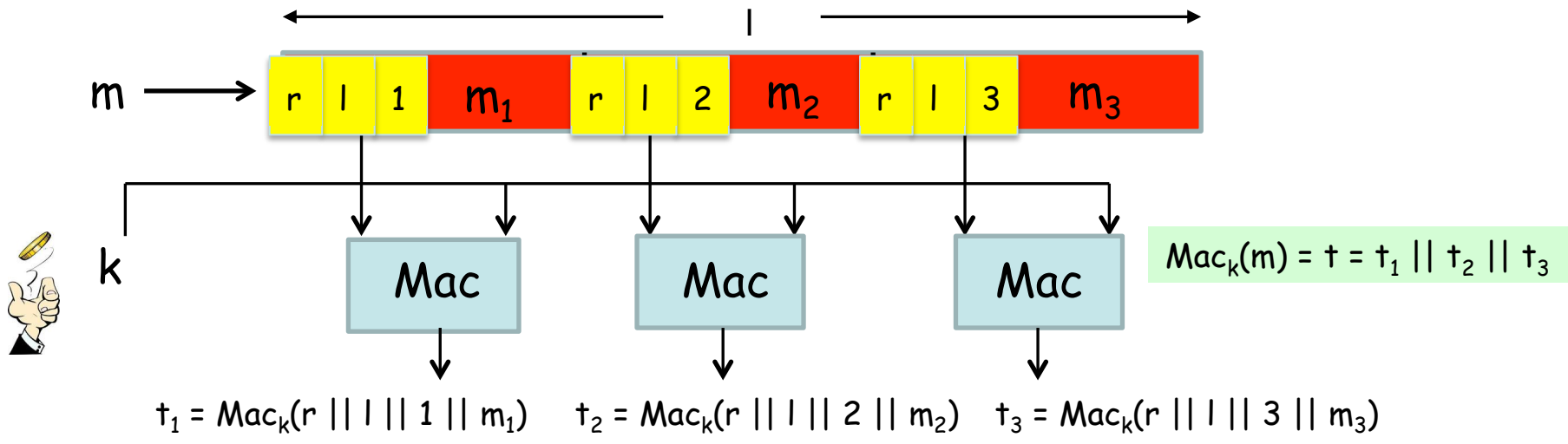
Domain Extension for MAC

Ahhhh Finally!

...t work !!

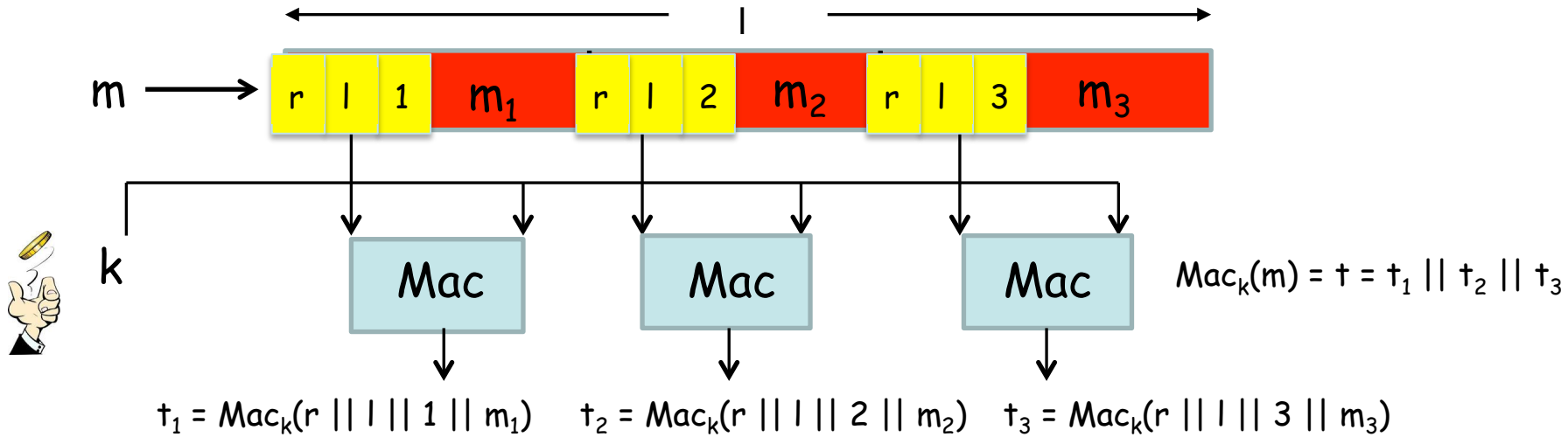
Attempt IV

- Prevent the previous attack by additionally authenticating a random identifier with each block; a fresh random identifier for each message



- Is this construction secure ? --- yes (it is in fact a randomized MAC)
- Is Randomization necessary for domain extension?-- NO
- But this is highly inefficient --- each invocation of Mac is now invoked only on n/4 bits of m
  - So if |m| = dn bits, then it requires 4d invocations of Mac algorithm and tag size is 4dn bits

Proof of Domain Extension for MAC



Theorem: If Π' = (Mac', Vrfy') is CMA-secure for fixed-length message of length n, then Π = (Mac, Vrfy) is CMA-secure for arbitrary -length messages.

Proof: On the board.

CBC-MAC for Arbitrary-length Messages

- Let F: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a PRF, whose key k is agreed between S and R
- Let S has a message m with $|m| = dn$, where d is some polynomial in n
- CBC-Mac:

Practical Domain Extension: CBC MAC & Proof & Differences with CBC Mode of operation for SKE.

3rd Chalk and Talk topic

Information-theoretic MAC (no assumption, simple construction, strong security, very useful in high-level problems)

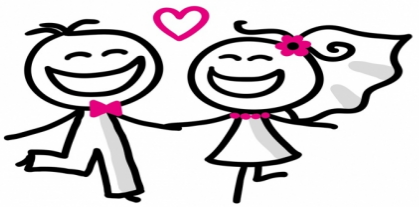
4th Chalk and Talk topic

- Only d invocations of PRF

Highly efficient

4d invocations of PRF

The Picture Till Now



SKE

- Privacy
- Not necessarily provide integrity and authentication;

>> easy to come of with a valid ciphertext

>> easy to manipulate known ciphertext

MAC

- Integrity & Authentication
- Not necessarily provide privacy;

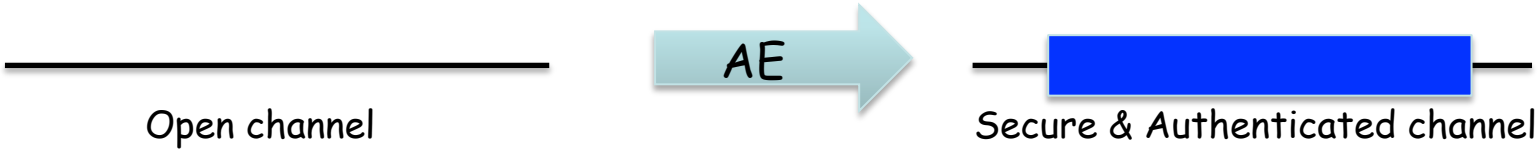
>> Easy to distinguish tags of two different messages

Authenticated Encryption

Jonathan Katz, Moti Yung:
Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation. FSE 2000: 284-299

Mihir Bellare, Chanathip Namprempre:
Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. ASIACRYPT 2000: 531-545

Authenticated Encryption



- But how do we define such
- Way out: try to captur in the definition
- Let Π = (Gen, Enc, D nd the following secrecy and integri an AE scheme :

Π is an authenticated encryption scheme if no PPT attacker is able to non-negligibly win the CCA-experiment and Enc-Forge experiment with respect to Π

  - For secrecy, we demand CCA s rity: no PPT attacker should be able to non-negligibly distinguish betwee cryption of two mess of it has access to encryption and decrypti hope for at the privacy front

  - For integrity/authentication, we d MAC. No PPT attacker who might hav in the "past" is unable to come up with a vali ciphertext) for to a (new) message for which he has never se ciphertext.

    - Modeled via a new experiment which exactly captures the above --- Enc-Forge

>> Enc-Forge is similar in spirit of Mac-forge

>> We need to introduce new game and definition since MAC and SKE has different sintax

Unforgeable Encryption Experiment

Experiment Enc-Forge$_{A,\Pi}$(n)

Π = (Gen, Enc, Dec)

PPT Attacker A

I can forge Π

$Q = \{m_1, \ldots, m_t\}$

Encryption Oracle

message

Encryption

Ciphertext c

Let me verify

k

Gen(1^n)

game output

Deck(c) = m ≠ ⊥ and $m \notin Q$ → 1

Deck(c) = m = ⊥ or $m \in Q$ → 0

Π is unforgeable if for every PPT A:

$$\Pr\left[\text{Enc-Forge}_{A,\Pi}(n) = 1\right] \leq \text{negl}(n)$$

Authenticated Encryption (Formal Definition)

- A symmetric-key cipher Π = (Gen, Enc, Dec) is an authenticated cipher if both the following holds:
  - Π is CCA-secure
    - For every PPT adversary A participating in the CCA-experiment, there is a negligible function $negl_1()$, such that:

$$\Pr\left[\mathrm{PrivK}^{cca}_{A,\Pi}(n) = 1\right] \leq \frac{1}{2} + negl_1(n)$$

  - Π is unforgeable
    - For every PPT adversary A participating in the unforgeable encryption experiment, there is a negligible function $negl_2()$, such that:

$$\Pr\left[\text{Enc-Forge}_{A,\Pi}(n)\right] \leq negl_2(n)$$

Thank You!

CBC-MAC vs CBC-mode of Encryption

- Random IV present in CBC-mode of encryption
  - Very crucial for security
- Will there be any harm if we use a random IV in CBC-MAC ?
  - Yes; it will become insecure !!
- In CBC-mode of encryption, the intermediate values are also part of the output (ciphertext)
- Will there be any harm if we include the intermediate values in CBC-MAC as part of the tag ?
  - Yes; it will become insecure !!
- We should be very careful in implementing crypto primitives
  - Should clearly follow the specifications