期中 11月8日

Pset 2 non-uniform

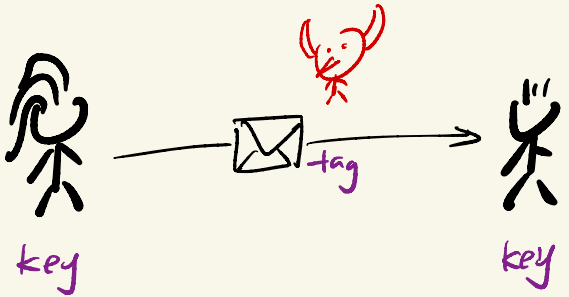
Pset 3 $\varepsilon_i \in negl(n) \not\Rightarrow \varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_n \in negl(n)$

$\exists i \ \varepsilon_i \notin negl(n) \not\Rightarrow \varepsilon_1 + \varepsilon_2 \notin negl(n)$

Chosen-ciphertext attack (CCA)

message authentication code (MAC)

- Secrecy
- Integrity



Previous:
Passive Attack

Today:
Active Attack

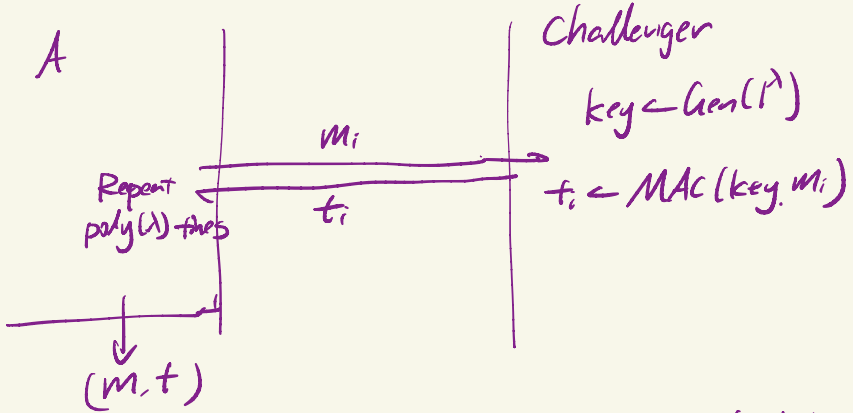
MAC

- Key generation Gen $(1^\lambda) \to$ key
- Tag generation MAC (key, msg) $\to$ tag
- Verification Verify (key, msg, tag) $\to$ {Acc, Rej}
  T F

Correctness:

key $\leftarrow$ Gen (1^λ)
tag $\leftarrow$ MAC (key, msg)
Acc $\leftarrow$ Verify (key, msg, tag)

MAC (strong) Forgeability Game $\text{MAC-forge}^{s}_{A,\Pi}(1^\lambda)$



A wins if and only if
- ① $\text{Verify}(\text{key}, m, t) \to \text{Acc}$
- (2a) $(m,t) \neq (m_i, t_i)\ \forall i$
- (2b) $m \neq m_i\ \forall i$

Π is
- unforgeable if $\forall$ p.p.t. A $\quad \Pr[\text{MAC-forge}_{A,\Pi}(1^\lambda) \to 1] \leq \text{negl}(\lambda)$
- strongly unforgeable if $\forall$ p.p.t. A $\quad \Pr[\text{MACSforge}_{A,\Pi}(1^\lambda) \to 1] \leq \text{negl}(\lambda)$

MAC needs randomness?

NO. deterministic MAC.

⇓

$\text{Verify}(k, m, t)$ = check if $t = \text{MAC}(k, m)$

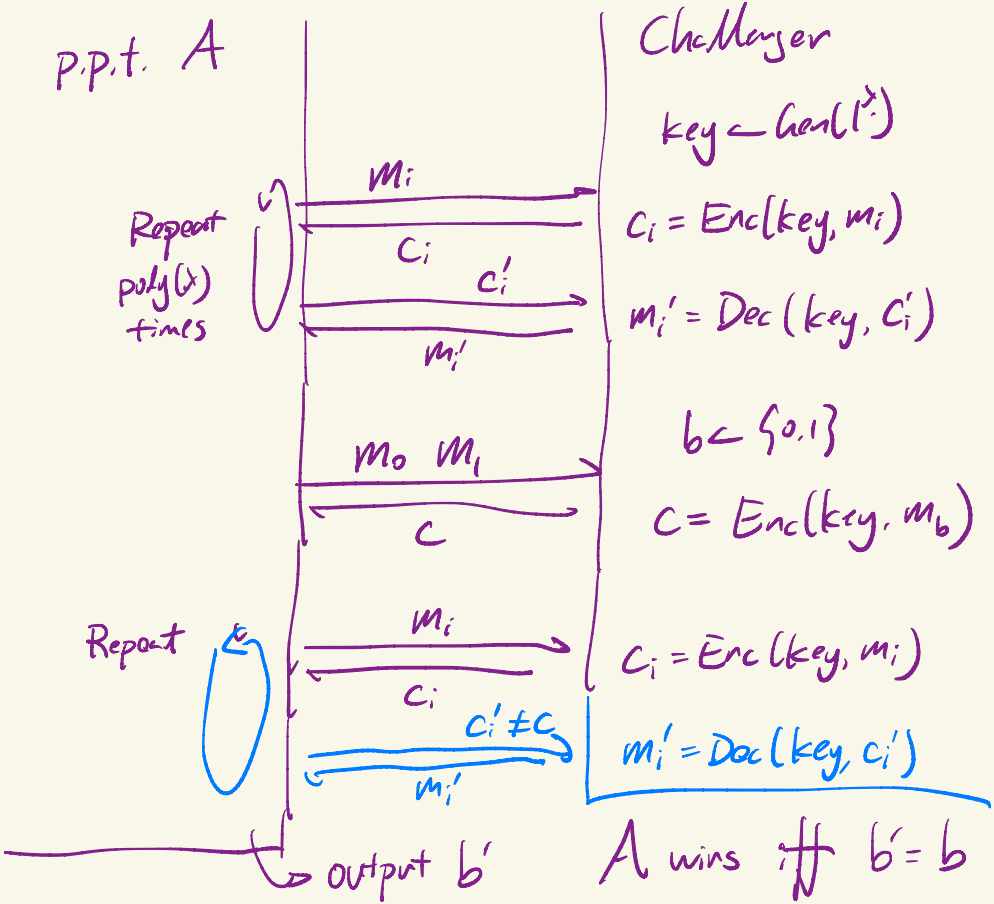
⇓

unforgeability ≡ strongly unforgeability

Why No verification oracle?

Chosen-Ciphertext Attack

$PrivK^{CCA1}_{A,\Pi}(\lambda)$ $PrivK^{CCA2}_{A,\Pi}(\lambda)$



CPA-secure encryption scheme + Unforgeable MAC $\Rightarrow$ CCA1-secure

Enc
Dec
.... + strongly unforgeable MAC $\Rightarrow$ CCA2-secure

$\Rightarrow$ Authenticated Encryption

Candidate (MAC-then-encrypt)

$$Enc'_{k,k}(m) = Enc_k(m, MAC_k(m))$$

$$Dec'_{k,k}(c) = m,t \leftarrow Dec_k(c)$$

$Verify_k(m,t)$ accepts $\Rightarrow$ output m

otherwise $\Rightarrow$ output $\perp$

Candidate (encrypt-then-MAC)

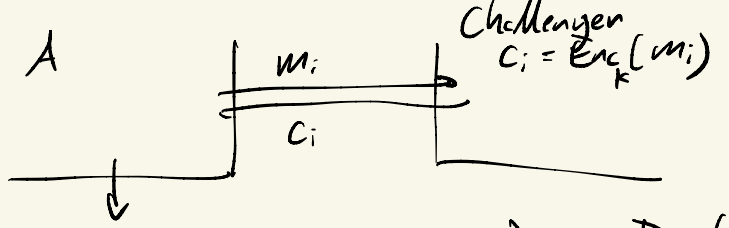
$$Enc'_{k,k}(m) = \left(c = Enc_k(m),\ t = MAC_k(c)\right)$$

$$Dec'_{k,k}(c,t) = Verify_k(c,t)$$ reject $\Rightarrow$ output $\perp$

otherwise $\Rightarrow$ $Dec_k(c)$

Def. Authenticated Encryption = CCA + Unforgeable

Encryption Unforgeability Game $\text{Enc-Forge}_{A,\Pi}(\lambda)$



A wins iff $\begin{cases} m \leftarrow Dec_k(c). \\ \text{①} \quad m \neq \perp \\ \text{②} \quad m \neq m_i \ \forall i \end{cases}$

Constructions of MAC

MAC for fixed-length messages. assume PRF $f: \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^\lambda$

- $Gen(1^\lambda)$ sample $k \leftarrow \{0,1\}^\lambda$
- $MAC(k, m) = f_k(m)$
- $Verify(k, m, t) = [f_k(m) = t]$

- $MAC(k, (m_1, m_2, m_3 \cdots m_n))$

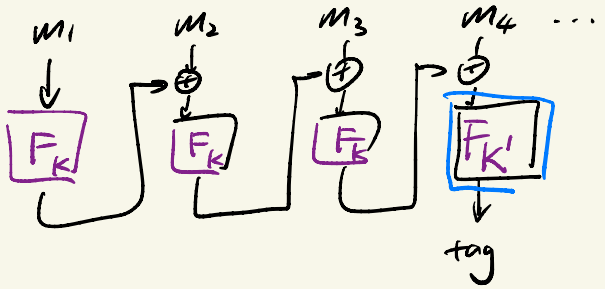
$$\left(2V \xleftarrow{\$}, \; f_k(IV \| m_i \| i \| n) \text{ for } i = 1, \cdots, n \right)$$

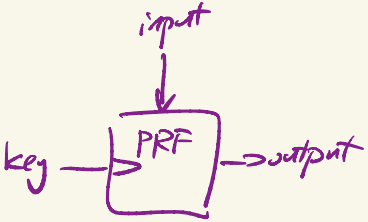
$m_1\ m_2 \xrightarrow{MAC} t_1\ t_2$

$m_1'\ m_2' \xrightarrow{MAC} t_1'\ t_2'$

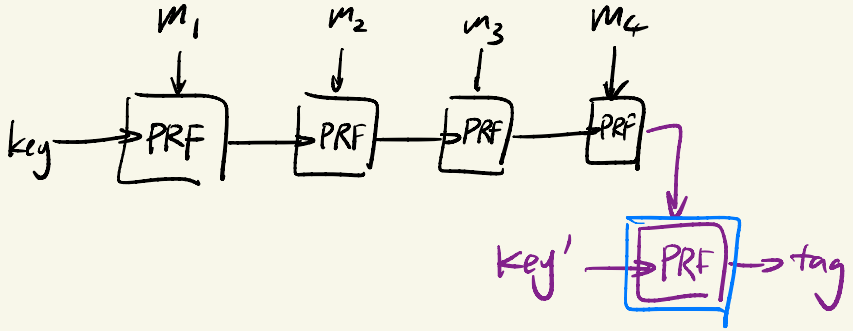
$m_1\ m_2' \xrightarrow{MAC} t_1\ t_2'$

CBC-mode

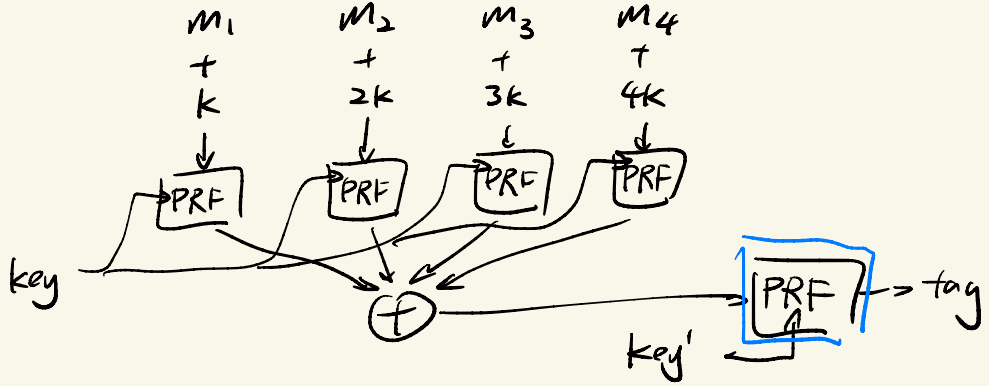




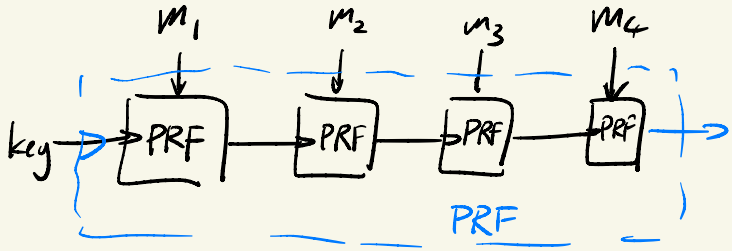
NMAC (cascade)



PMAC

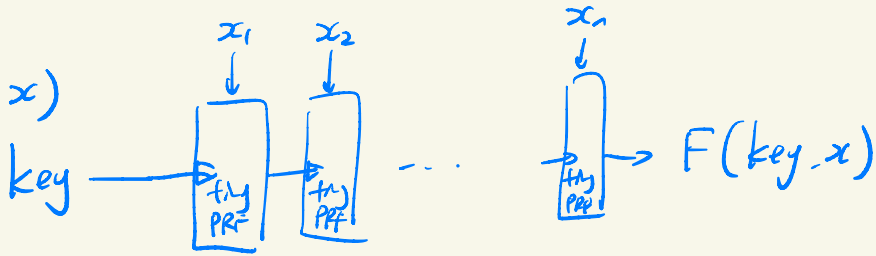


NMAC (cascade)



PRG ⇒ PRF

$F(key, x)$



$$\text{tiny PRF}: \{0,1\}^\lambda \times \{0,1\} \to \{0,1\}^\lambda$$

$$key, x \to \begin{cases} G(key)[1:\lambda] & \text{if } x=0 \\ G(key)[\lambda+1:2\lambda] & \text{if } x=1 \end{cases}$$

Prefix-free PRF

Prefix-free encoding : $E: \{0,1\}^* \to \{0,1\}^*$

$\forall x_1, x_2, \quad E(x_1)$ is not a prefix of $E(x_2)$

$$\text{Im } E = E(\{0,1\}^*) = \{ E(x) \mid x \in \{0,1\}^* \}$$

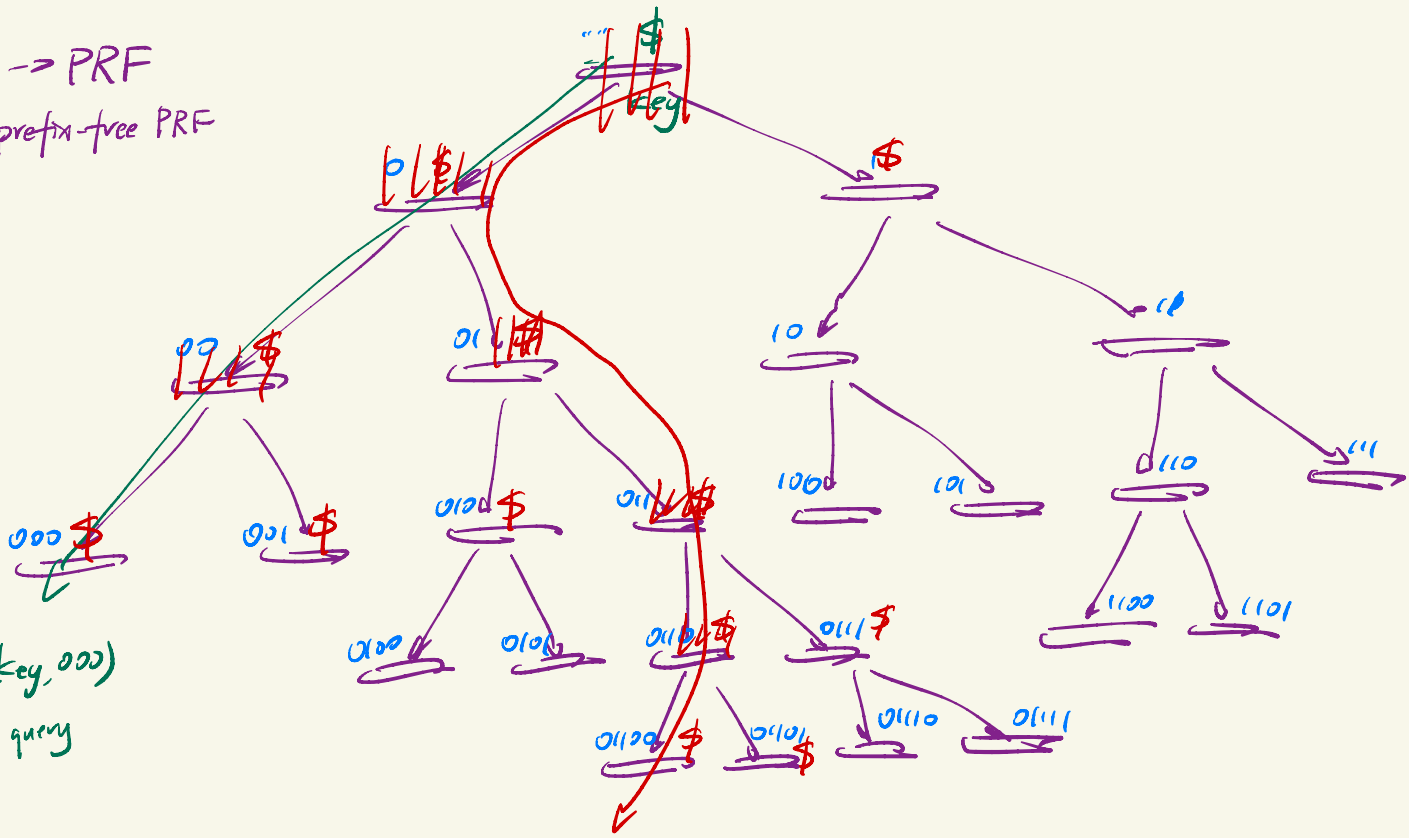
PRF for all strings $\iff$ prefix-free encoding + prefix-free PRF

$F^0: \{0,1\}^\lambda \times \{0,1\}^* \to \{0,1\}^\lambda$ $\qquad$ E $\qquad$ $F: \{0,1\}^\lambda \times \text{Im } E \to \{0,1\}^\lambda$

$F^0(x) = F(\text{key}, E(x))$

PRG -> PRF

↳ prefix-free PRF



F(key, 000)

First query

F(key, 01100)

second query

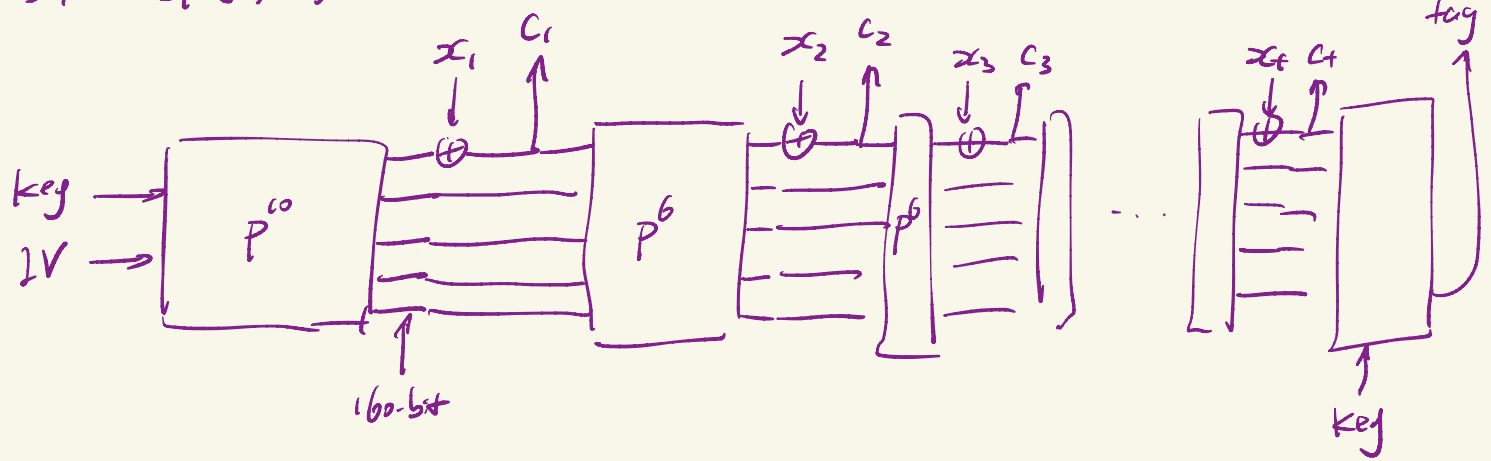
Practical Authenticated Encryption

E.g. ASCON cipher

key

IV

$x_1 \cdots x_t \in \{0,1\}^{32}$



$\underset{\text{text}}{\text{cipher}}(IV, c_1 \cdots c_t, tag)$