**CS655: Cryptography** **Feb 13, 2007**

Lecture Outline for Message Authentication

# 6 Message Authentication

Readings: Sections 6.1–6.7 of Bellare&Rogaway

- Beyond privacy.
- Message authentication = Data-origin authentication.
- Shared key vs. public key (digital signature).

## 6.1 The setting

- A message authentication scheme: each message $M$ is transformed with a key $K$ to $M'$.
- A message authentication code (MAC) scheme: the transformation involves adding a tag (MAC).
- To compromise authenticity, require adversary to to active attack.

## 6.2 Privacy does not imply authenticity

- While sending $E_K(M)$ does not provide authenticity? Without knowing the key, the attacker cannot encrypt the message s/he wants to send, right?
  - No. What about using one-time pad, or stream cipher, or block cipher with modes that behave in stream-lie modes (e.g., CTR)?
- Should conclude that encryption is not for authenticity.

## 6.3 Syntax of message-authentication schemes

- A *message authentication scheme* $\mathcal{MA} = (\mathcal{K}, \mathcal{S}, \mathcal{R})$.
- A *message authentication code* (MAC) scheme consists of three algorithms $\Pi = (\mathcal{K}, \mathsf{MAC}, \mathsf{VF})$.
  - $\mathcal{K}$ is a randomized key generation algorithm. $\mathsf{Keys}(()\Pi)$ denotes the set of all keys that have non-zero probability of being generated.
  - The MAC-generation algorithm $\mathsf{MAC}$ (might be randomized or stateful) takes a key $K \in \mathsf{Keys}(()\Pi)$ and a message $M \in \{0,1\}^*$ and return a tag $\mathit{Tag} \in \{0,1\}^* \cup \{\perp\}$.
  - The deterministic MAC-verification algorithm $\mathsf{VF}$ takes a key $K \in \mathsf{Keys}(()\Pi)$, a message $M \in \{0,1\}^*$, and a candidate tag $\mathit{Tag} \in \{0,1\}^*$, and returns either 1 (accept) or 0 (reject).
- An MAC scheme naturally gives rise to a message authentication scheme.
- For an encryption scheme to be secure, it must be either probabilistic or stateful. However, an MAC scheme can be deterministic and stateless, in which case the verification is typically done calculating the authentication tag again and comapre.

## 6.4 A definition of security for MACs

- Don't want the adversary to create a pair $(M, Tag)$ such that $\mathsf{VF}_K(M, Tag) = 1$, which is called a forgery.
- Chosen-message attack.
- Security definition. The experiment $\mathbf{Exp}^{\mathsf{uf\text{-}cma}}_{\Pi}(A)$.

  Note that the adversary is given two (related) oracles: $\mathsf{MAC}_K(\cdot)$ and $\mathsf{VF}(\cdot)$.

## 6.5 Examples

- Given PRF $F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$. Suppose all messages are of length $\ell$, then a secure MAC scheme is $Tag = F_K(M)$.

  How to construct MAC for long messages?
- **Insecure MAC Construction 1:** Divide message into blocks of length $\ell$: $M = M[1] \cdots M[n]$, then $Tag = F_K(M[1]) \oplus \cdots \oplus F_K(M[n])$. How to break the security?
- **Insecure MAC Construction 2:** Divide message into blocks of length $t = \ell - m$: $M = M[1] \cdots M[n]$, then $Tag = F_K([i]_m \| M[i])$. How to break this?

## 6.6 The PRF-as-a-MAC Paradigm

## 6.7 The CBC MACs

- **Basic CBC MAC** Given a block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. Divide $M$ into $n$-bit blocks $M[1] \cdots M[m]$. Let $C[0] = 0^n$, and $C[i] = E_K(C[i-1] \oplus M[i]$ for $1 \le i \le m$. Let $Tag = C[m]$.

  This is secure if all messages are of the same length, but insecure otherwise.