A Quick Tour of Cryptographic Primitives

Anupam Datta
CMU
Fall 2016
Basic Cryptographic Concepts

- Encryption scheme (symmetric and public key)
- Signature scheme
- Message authentication code
- Hash function
Symmetric Encryption Scheme

- **Key generation algorithm**
  - Produces a key that is used for encryption and decryption

- **Algorithm to encrypt a message**

- **Algorithm to decrypt a ciphertext**

- **Correctness:**
  - Decrypting a ciphertext obtained by encrypting message \( m \) with the corresponding key \( k \) returns \( m \)
  \[
  \text{dec(\text{enc}(m,k),k)} = m
  \]

- **(Symbolic) Security:**
  - A ciphertext cannot be decrypted without access to the key

*Can you think of a stronger security property?*
Public-Key Encryption Scheme

- **Key generation algorithm**
  - Produces private decryption & public encryption key pair

- Algorithm to *encrypt* a message

- Algorithm to *decrypt* a ciphertext

- **Correctness:**
  - Decrypting a ciphertext obtained by encrypting message \( m \) with the corresponding encryption key returns \( m \)
    
    \[
    \text{dec}(\text{enc}(m, pk(A)), sk(A)) = m
    \]

- **(Symbolic) Security:**
  - A ciphertext cannot be decrypted without access to the private decryption key

---

*Why would you want public key encryption?*
Signature Scheme

- **Key generation algorithm**
  - Produces private signing & public verification key pair
- Algorithm to **sign** data
- Algorithm to **verify** signature
- **Correctness:**
  - Message signed with a signing key verifies with the corresponding verification key
    \[
    \text{verify}(m, \text{sign}(m, \text{sk}(A)), \text{pk}(A)) = \text{ok}
    \]
- **Security:**
  - A signature cannot be produced without access to the private signing key

---

**Can you think of a scenario where you may not want non-repudiation?**
Message Authentication Code (MAC)

- Key generation algorithm
  - Produces a key
- Algorithm to \textit{mac} a message
- Algorithm to \textit{verify} a mac on a message
- Correctness:
  - Message mac-ed with key verifies with the same key
    \[
    \text{verify}(k, m, \text{mac}(k,m)) = \text{ok}
    \]
- Security:
  - A MAC cannot be produced without access to the key

Similar to signature, but uses symmetric key

\textit{What property does a signature have, but a MAC does not?}
Hash Functions

- **Key generation algorithm**
  - Produces a key
- **Algorithm to hash a message m, given a key k to a fixed length output** $\text{hash}(k, m)$

- **Security (Collision resistance)**

Given hash function $\text{hash}: X \rightarrow Y$ and key $k$, cannot find a collision, i.e. $x, x' \in X$ s.t. $x \neq x'$ and $\text{hash}(k,x) = \text{hash}(k,x')$

**What is the difference between a MAC and a hash function?**
Cryptographic Constructions

- **Signature scheme**
  - DSS, RSA-FDH,…

- **Hash function**
  - SHA-1, MD5,…

- **Message Authentication Code (MAC)**
  - HMAC, CBC-MAC, NBAC,…

- **Encryption scheme**
  - Asymmetric (public key): RSA, Diffie-Hellman, El Gamal,… (distinct keys for encryption and decryption)
  - Symmetric: DES, 3DES, AES, RC4,… (same key for encryption and decryption)

Want to know more? 18-733: Applied Cryptography