CRYPTO REVIEW

SHIQING MA

RUTGERS CS 419

GOALS OF CRYPTOGRAPHY

- The most fundamental problem cryptography addresses: ensure security of communication over insecure medium
- What does secure communication mean?
  - confidentiality (privacy, secrecy)
    - only the intended recipient can see the communication
  - integrity (authenticity)
    - the communication is generated by the alleged sender
- What does insecure medium mean?
  - Two possibilities:
    - Passive attacker: the adversary can eavesdrop
    - Active attacker: the adversary has full control over the communication channel

SHIFT CIPHER

- The Key Space:
  - [0 .. 25]
- Encryption given a key K:
  - each letter in the plaintext P is replaced with the K'th letter following corresponding number (shift right)
- Decryption given K:
  - shift left

History: K = 3, Caesar's cipher

HOW TO DEFEAT FREQUENCY ANALYSIS?

- Use larger blocks as the basis of substitution. Rather than substituting one letter at a time, substitute 64 bits at a time, or 128 bits.
  - Leads to block ciphers such as DES & AES.

- Use different substitutions to get rid of frequency features.
  - Leads to polyalphabetical substituion ciphers
  - Stream ciphers

ADVERSARIAL MODELS FOR CIPHERS

- The language of the plaintext and the nature of the cipher are assumed to be known to the adversary.
- **Ciphertext-only attack:** The adversary knows only a number of ciphertexts.
- **Known-plaintext attack:** The adversary knows some pairs of ciphertext and corresponding plaintext.
- **Chosen-plaintext attack:** The adversary can choose a number of messages and obtain the ciphertexts
- **Chosen-ciphertext attack:** The adversary can choose a number of ciphertexts and obtain the plaintexts.

What kinds of attacks have we considered so far?

When would these attacks be relevant in wireless communications?

ONE-TIME PAD

Let $Z_m = \{0,1,\ldots,m-1\}$ be the alphabet.

Plaintext space = Ciphtertext space = Key space $= (Z_m)^n$

The key is chosen uniformly randomly

Plaintext $X = (x_1\ x_2\ \ldots\ x_n)$

Key $K = (k_1\ k_2\ \ldots\ k_n)$

Ciphertext $Y = (y_1\ y_2\ \ldots\ y_n)$

$e_k(X) = (x_1+k_1\ \ x_2+k_2\ \ldots\ x_n+k_n) \bmod m$

$d_k(Y) = (y_1-k_1\ \ \ y_2-k_2\ \ldots\ \ y_n-k_n) \bmod m$

SHANNON (INFORMATION-THEORETIC) SECURITY = PERFECT SECRECY

Basic Idea: Ciphertext should reveal no "information" about plaintext

Definition. An encryption over a message space $\mathcal{M}$ is perfectly secure if

$\forall$ probability distribution over $\mathcal{M}$

$\forall$ message $m \in \mathcal{M}$

$\forall$ ciphertext $c \in \mathcal{C}$ for which $Pr[C=c] > 0$

We have

$$Pr[\mathbf{PT}=m \mid \mathbf{CT}=c] = Pr[\mathbf{PT} = m].$$

STREAM CIPHERS

- In One-Time Pad, a key is a random string of length at least the same as the message
- Stream ciphers:
  - Idea: replace "rand" by "pseudo rand"
  - Use Pseudo Random Number Generator
  - PRNG: $\{0,1\}^s \rightarrow \{0,1\}^n$
    - expand a short (e.g., 128-bit) random seed into a long (e.g., 10^6 bit) string that "looks random"
  - Secret key is the seed
  - $E_{key}[M] = M \oplus PRNG(key)$

TOWARDS COMPUTATIONAL SECURITY

- Perfect secrecy is too difficult to achieve.
- The computational approach uses two relaxations:
  - Security is preserved only against **efficient** (computationally bounded) adversaries
    - Adversary can only run in feasible amount of time
  - Adversaries can potentially succeed with some **very small probability** (that we can ignore the case it actually happens)
- Two approaches to formalize computational security: concrete and asymptotic

TOWARDS IND-CPA SECURITY:

- Ciphertext Indistinguishability under a Chosen-Plaintext Attack: Define the following IND-CPA experiment :
  - Involving an Adversary and a Challenger
  - Instantiated with an Adversary algorithm A, and an encryption scheme Π = (Gen, Enc, Dec)

WHY BLOCK CIPHERS?

- One thread of defeating frequency analysis
  - Use different keys in different locations
  - Example: one-time pad, stream ciphers

- Another way to defeat frequency analysis
  - Make the unit of transformation larger, rather than encrypting letter by letter, encrypting block by block
  - Example: block cipher

BLOCK CIPHER ENCRYPTION MODES: ECB

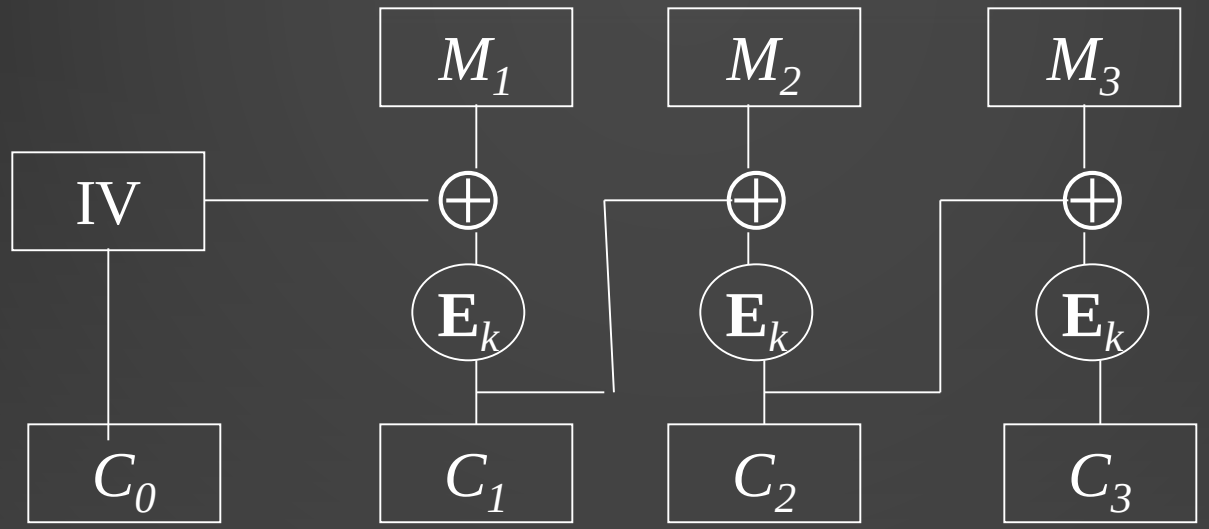
- Message is broken into independent blocks;
- Electronic Code Book (ECB): each block encrypted separately.
- **Encryption: $c_i = E_k(x_i)$**
- **Decrytion: $x_i = D_k(c_i)$**

DES ENCRYPTION MODES: CBC

- Cipher Block Chaining (CBC):
  - Uses a random Initial Vector (IV)
  - Next input depends upon previous output

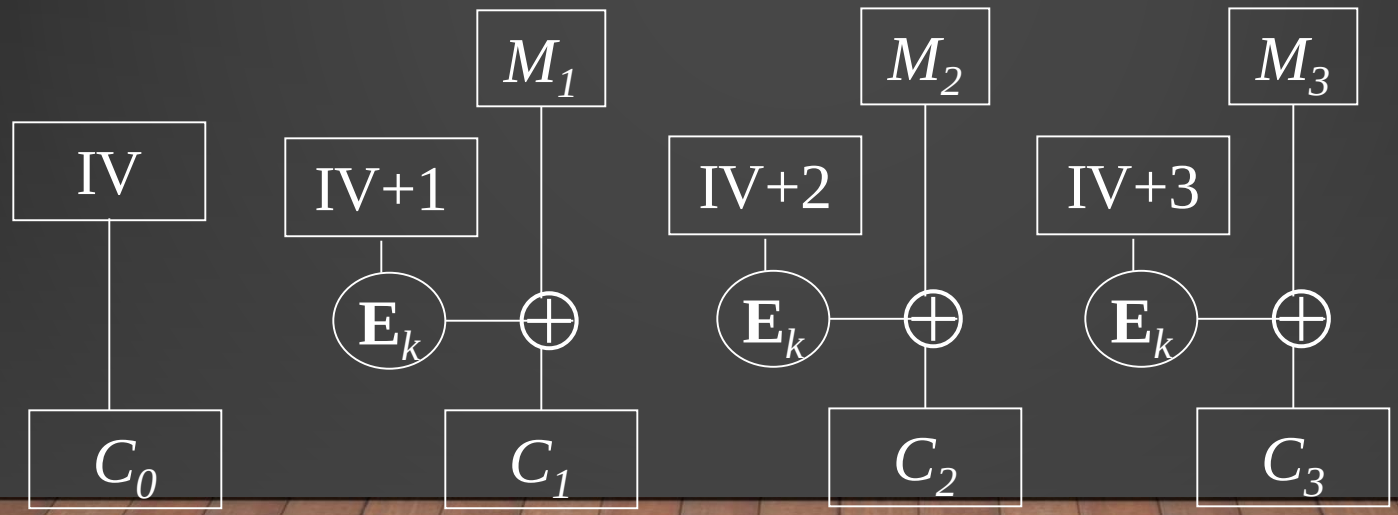
Encryption: $C_i = E_k(M_i \oplus C_{i-1})$, **with** C_0=**IV**

Decryption: $M_i = C_{i-1} \oplus D_k(C_i)$, **with** C_0=**IV**

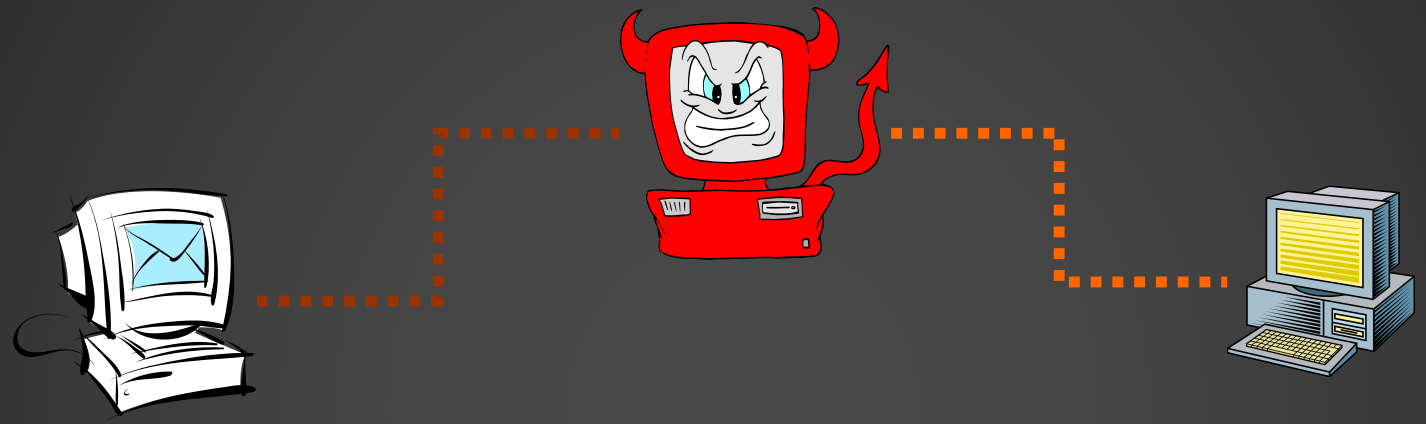


ENCRYPTION MODES: CTR

- Counter Mode (CTR): Defines a stream cipher using a block cipher
  - Uses a random IV, known as the counter
  - Encryption: C_0=IV, $C_i = M_i \oplus E_k[IV+i]$
  - Decryption: IV=C_0, $M_i = C_i \oplus E_k[IV+i]$



INTEGRITY AND AUTHENTICATION



- Encryption does not protect data from modification by another party.
  - Why?
- Need a way to ensure that data arrives at destination in its original form as sent by the sender and it is coming from an authenticated source.

LIMITATION OF USING HASH FUNCTIONS FOR AUTHENTICATION

- Require an authentic channel to transmit the hash of a message
  - Without such a channel, it is insecure, because anyone can compute the hash value of any message, as the hash function is public
  - Such a channel may not always exist
- How to address this?
  - use more than one hash functions
  - use a key to select which one to use

MESSAGE AUTHENTICATION CODE

- A MAC scheme is a hash family, used for message authentication
- MAC(K,M) = $H_K(M)$
- The sender and the receiver share secret K
- The sender sends (M, $H_k(M)$)
- The receiver receives (X,Y) and verifies that $H_K(X)=Y$, if so, then accepts the message as from the sender
- To be secure, an adversary shouldn't be able to come up with (X',Y') such that $H_K(X')=Y'$.

HMAC: CONSTRUCTING MAC FROM CRYPTOGRAPHIC HASH FUNCTIONS

$$HMAC_K[M] = Hash[(K^+ \oplus opad) \,||\, Hash[(K^+ \oplus ipad)||M)]]$$

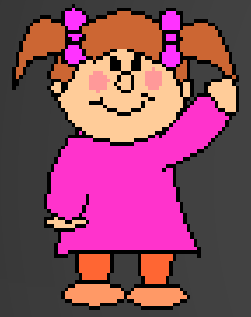
- K^+ is the key padded (with 0) to B bytes, the input block size of the hash function
- ipad = the byte 0x36 repeated B times
- opad = the byte 0x5C repeated B times.

At high level, $HMAC_K[M] = H(K \,||\, H(K \,||\, M))$

CONCEPT OF PUBLIC KEY ENCRYPTION

- Each party has a pair (K, K^{-1}) of keys:
  - K is the **public** key, and used for encryption
  - K^{-1} is the **private** key, and used for decryption
  - Satisfies $\mathbf{D}_{K^{-1}}[\mathbf{E}_K[M]] = M$
- Knowing the public-key K, it is computationally infeasible to compute the private key K^{-1}
  - How to check (K,K^{-1}) is a pair?
  - Offers only computational security. Secure PK Encryption impossible when P=NP, as deriving K^{-1} from K is in NP.
- The public-key K may be made publicly available, e.g., in a publicly available directory
  - Many can encrypt, only one can decrypt
- Public-key systems aka *asymmetric* crypto systems

DIFFIE-HELLMAN KEY AGREEMENT PROTOCOL



$g^a \bmod p$

$g^b \bmod p$

Pick random, secret a

Compute and send $g^a \bmod p$

$K = (g^b \bmod p)^a = g^{ab} \bmod p$

Pick random, secret b

Compute and send $g^b \bmod p$

$K = (g^a \bmod p)^b = g^{ab} \bmod p$

RSA PUBLIC KEY CRYPTO SYSTEM

Key generation:

1. Select 2 large prime numbers of about the same size, p and q
   Typically each p, q has between 512 and 2048 bits
2. Compute n = pq, and $\Phi(n) = (q-1)(p-1)$
3. Select e, $1<e<\Phi(n)$, s.t. $\gcd(e, \Phi(n)) = 1$
   Typically e=3 or e=65537
4. Compute d, $1< d<\Phi(n)$ s.t. $ed \equiv 1 \bmod \Phi(n)$
   Knowing $\Phi(n)$, d easy to compute.

Public key: (e, n)

Private key: d

DIGITAL SIGNATURES

- Digital Signature: a data string which associates a message with some originating entity.
- Digital Signature Scheme:
  - a signing algorithm: takes a message and a (private) signing key, outputs a signature
  - a verification algorithm: takes a (public) verification key, a message, and a signature
- Provides:
  - Authentication, Data integrity, Non-Repudiation

THE BIG PICTURE

Shiqing Ma, Rutgers University

	Secret Key Setting	Public Key Setting
Secrecy / Confidentiality	Stream ciphers Block ciphers + encryption modes	Public key encryption: RSA, El Gamal, etc.
Authenticity / Integrity	Message Authentication Code	Digital Signatures: RSA, DSA, etc.

NEEDHAM-SCHROEDER SHARED-KEY PROTOCOL

- Parties: A, B, and trusted server T
- Setup: A and T share K_{AT}, B and T share K_{BT}
- Goal: Mutual entity authentication between A and B; key establishment
- Messages:

$A \rightarrow T$: A, B, N_A (1)

$A \leftarrow T$: $E[K_{AT}]\ (N_A, B,\ k,\ E[K_{BT}](k,A))$ (2)

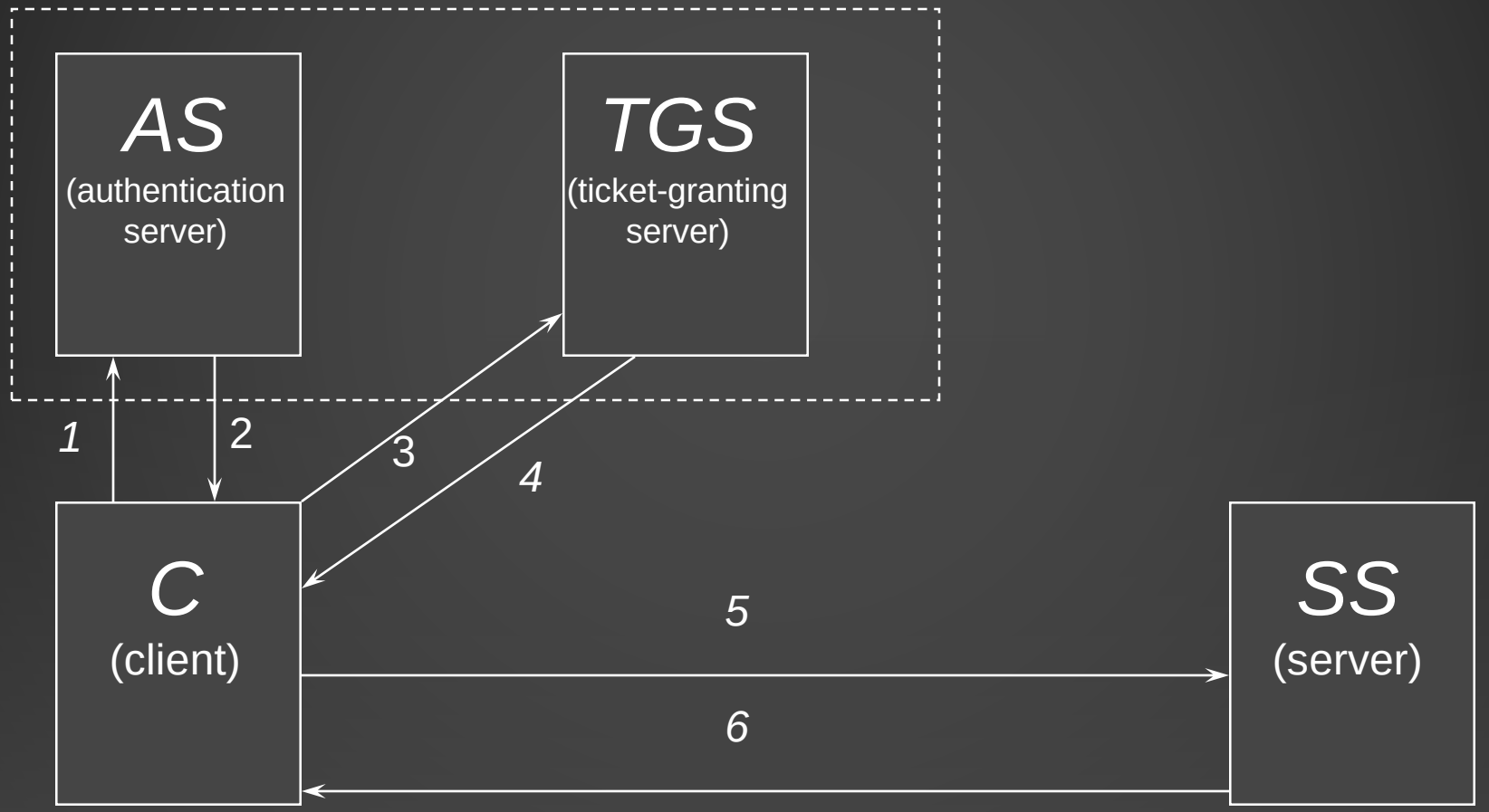
$A \rightarrow B$: $E[K_{BT}]\ (k, A)$ (3)

$A \leftarrow B$: $E[k]\ (N_B)$ (4)

$A \rightarrow B$: $E[k]\ (N_B\text{-}1)$ (5)

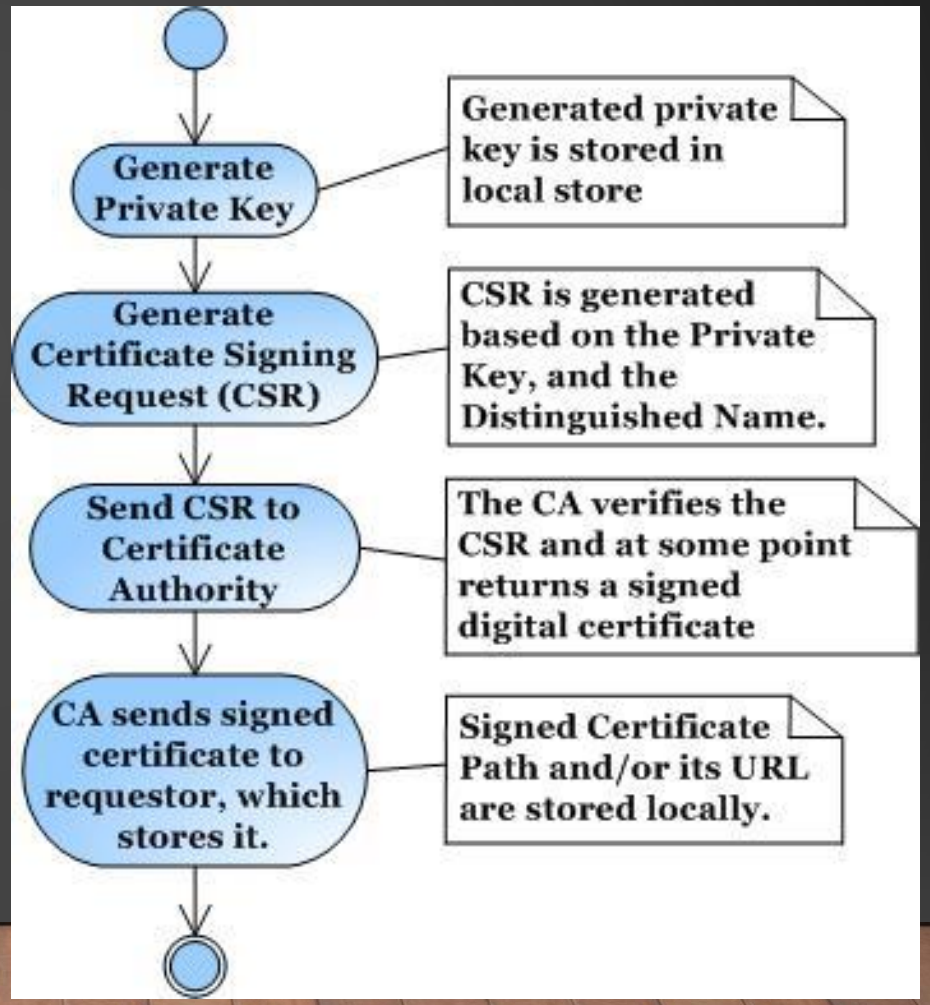
What bad things can happen if there is no NA.

KERBEROS PROTOCOL - 1



HOW TO OBTAIN A CERTIFICATE?

- Define your own CA (use openssl or Java Keytool)
  - Certificates unlikely to be accepted by others
- Obtain certificates from one of the vendors: VeriSign, Thawte, and many others



QUIZ TIME