

University of Rochester
CSC290B
Introduction to Computer Security

Key Management

March 31, 2009

Public-Key Cryptosystems: Security Services

- Confidentiality
 - Only the owner of the private key knows it, so text enciphered with public key cannot be read by anyone except the owner of the private key
- Authentication
 - Only the owner of the private key knows it, so text enciphered with private key must have been generated by the owner

Public-Key Cryptosystems: More Security Services

- Integrity
 - Enciphered letters cannot be changed undetectably without knowing private key
- Non-Repudiation
 - Message enciphered with private key came from someone who knew it

Public-Key Cryptosystems: Warnings

- Encipher message in blocks considerably larger than the examples here
 - If 1 character per block, RSA can be broken using statistical attacks (just like classical cryptosystems)
 - Attacker cannot alter letters, but can rearrange them and alter message meaning
 - Example: reverse enciphered message of text ON to get NO

Cryptographic Checksums

- Mathematical function to generate a set of k bits from a set of n bits (where $k \leq n$).
 - k is smaller than n except in unusual circumstances
- Example: ASCII parity bit
 - ASCII has 7 bits; 8th bit is “parity”
 - Even parity: even number of 1 bits
 - Odd parity: odd number of 1 bits

Example Use

- Bob receives “10111101” as bits.
 - Sender is using even parity; 6 1 bits, so character was received correctly
 - Note: could be garbled, but 2 bits would need to have been changed to preserve parity
 - Sender is using odd parity; even number of 1 bits, so character was not received correctly

Definition

- Cryptographic checksum $h: A \rightarrow B$:
 1. For any $x \in A$, $h(x)$ is easy to compute
 2. For any $y \in B$, it is computationally infeasible to find $x \in A$ such that $h(x) = y$
 3. It is computationally infeasible to find two inputs $x, x' \in A$ such that $x \neq x'$ and $h(x) = h(x')$
 - Alternate form (stronger): Given any $x \in A$, it is computationally infeasible to find a different $x' \in A$ such that $h(x) = h(x')$.

Collisions

- If $x \neq x'$ and $h(x) = h(x')$, x and x' are a *collision*
 - Pigeonhole principle: if there are n containers for $n+1$ objects, then at least one container will have 2 objects in it.
 - Application: if there are 32 files and 8 possible cryptographic checksum values, at least one value corresponds to at least 4 files

Keys

- Keyed cryptographic checksum: requires cryptographic key
 - DES in chaining mode: encipher message, use last n bits. Requires a key to encipher, so it is a keyed cryptographic checksum.
- Keyless cryptographic checksum: requires no cryptographic key
 - MD5 and SHA-1 are best known; others include MD4, HAVAL, and Snefru

HMAC

- Make keyed cryptographic checksums from keyless cryptographic checksums
- h keyless cryptographic checksum function that takes data in blocks of b bytes and outputs blocks of l bytes. k' is cryptographic key of length b bytes
 - If short, pad with 0 bytes; if long, hash to length b
- $ipad$ is 00110110 repeated b times
- $opad$ is 01011100 repeated b times
- $HMAC-h(k, m) = h(k' \oplus opad \parallel h(k' \oplus ipad \parallel m))$
 - \oplus exclusive or, \parallel concatenation
- This construction avoids “extension attacks” – simplest bad example is that a keyed hash $\equiv h(k \parallel m)$ would allow anyone to compute $h(k \parallel m \parallel anything)$ without knowing k .

Topics in Key Management

- Session and Interchange Keys
- Key Exchange
- Key Generation
- Cryptographic Key Infrastructure
- Storing and Revoking Keys
- Digital Signatures

Overview

- Key exchange
 - Session vs. interchange keys
 - Classical, public key methods
 - Key generation
- Cryptographic key infrastructure
 - Certificates
- Key storage
 - Key escrow
 - Key revocation
- Digital signatures

Notation

- $X \rightarrow Y : \{ Z \parallel W \} k_{X,Y}$
 - X sends Y the message produced by concatenating Z and W enciphered by key $k_{X,Y}$, which is shared by users X and Y
- $A \rightarrow T : \{ Z \} k_A \parallel \{ W \} k_{A,T}$
 - A sends T a message consisting of the concatenation of Z enciphered using k_A , A 's key, and W enciphered using $k_{A,T}$, the key shared by A and T
- r_1, r_2 nonces (nonrepeating random numbers)

Session, Interchange Keys

- Alice wants to send a message m to Bob
 - Assume public key encryption
 - Alice generates a random cryptographic key k_s and uses it to encipher m
 - To be used for this message *only*
 - Called a *session key*
 - She enciphers k_s with Bob;s public key k_B
 - k_B enciphers all session keys Alice uses to communicate with Bob
 - Called an interchange *key*
 - Alice sends $\{ m \}_{k_s} \{ k_s \}_{k_B}$

Benefits

- Limits amount of traffic enciphered with single key
 - Standard practice, to decrease the amount of traffic an attacker can obtain
- Prevents some attacks
 - Example: Alice will send Bob message that is either “BUY” or “SELL”. Eve computes possible ciphertexts $\{ \text{“BUY”} \} k_B$ and $\{ \text{“SELL”} \} k_B$. Eve intercepts enciphered message, compares, and gets plaintext at once

Acknowledgements

- Substantial portions of these slides are ©2002-2004 Matt Bishop and used with permission