{ CSCI 6331 · 4331 | Lecture 7 }

Cryptography

Hoeteck Wee · hoeteck@gwu.edu

http://tinyurl.com/cryptogw/

Announcements

- Evaluation:

  10% In-Class/Piazza, 20% Final Presentation / Project

  30% Homework, 40% Final (Apr 25)

- Homework 4?

"Secure Channels"

Q. How to guarantee both privacy and message integrity?

- SSL: protect passwords, credit card numbers between Web clients and servers
- IPSec: secure channel between two IP entities for protecting information at network layer
- SSH: remote logins

"Secure Channels"

Q. How to guarantee both privacy and message integrity?

Idea. Use encryption + MAC

- encryption: privacy
- MAC: message integrity (data origin authentication and data integrity)

Basic Principle. different security goals should always use independent keys

- k_1 encryption key, k_2 MAC key

Case study. EMV for credit and debit card payments

- a single key-pair is used for both signature and encryption
- Degabriele et al. 2012: theoretical attack for completing an offline transaction without knowing the cardholder's PIN

"Secure Channels"

- k_1 encryption key, k_2 MAC key

Encrypt and Authenticate. E&A in SSH:

— $\mathrm{c} \leftarrow \mathrm{Enc}_{\mathrm{k}_1}(\mathrm{m})$; $\mathrm{t} \leftarrow \mathrm{Mac}_{\mathrm{k}_2}(\mathrm{m})$; send $\mathrm{c}||\mathrm{t}$

— decrypt c to get m; verify tag t on m

Authenticate then Encrypt. AtE in SSL

— $\mathrm{t} \leftarrow \mathrm{Mac}_{\mathrm{k}_2}(\mathrm{m})$; $\mathrm{c} \leftarrow \mathrm{Enc}_{\mathrm{k}_1}(\mathrm{m}||\mathrm{t})$; send c

— decrypt c to get m, t; verify tag t on m

Encrypt then Authenticate. EtA in IPsec

— $\mathrm{c} \leftarrow \mathrm{Enc}_{\mathrm{k}_1}(\mathrm{m})$; $\mathrm{t} \leftarrow \mathrm{Mac}_{\mathrm{k}_2}(\mathrm{c})$; send $\mathrm{c}||\mathrm{t}$

— verify tag t on c; decrypt c

“Secure Channels”

Encrypt and Authenticate. E&A in SSH:

- $\mathrm{c} \leftarrow \mathrm{Enc}_{\mathrm{k}_1}(\mathrm{m})$; $\mathrm{t} \leftarrow \mathrm{Mac}_{\mathrm{k}_2}(\mathrm{m})$; send $\mathrm{c}||\mathrm{t}$
- decrypt c to get m; verify tag t on m

Q. integrity?

Q. privacy?

"Secure Channels"

Authenticate then Encrypt. AtE in SSL

— $t \leftarrow \mathrm{Mac}_{k_2}(m)$; $c \leftarrow \mathrm{Enc}_{k_1}(m||t)$; send c

— decrypt c to get m, t; verify tag t on m

Q. integrity?

Q. privacy?

— may break privacy if adversary can find out whether a given ciphertext is valid!

- idea: start flipping bits in the ciphertext
- e.g. by interacting with an on-line entity & observing network messages that it produces
- error messages indicating that decryption failed? c.f. encrypted email, acks in protocols

"Secure Channels"

Encrypt then Authenticate. EtA in IPsec

- $c \leftarrow \mathrm{Enc}_{k_1}(m)$; $t \leftarrow \mathrm{Mac}_{k_2}(c)$; send $c||t$
- verify tag t on c; decrypt c

Q. integrity?

Q. privacy?

- achieves privacy even if adversary can find out whether a given ciphertext is valid!

Deploying Private-Key Cryptography

Setting. Large organization with U users

- all pairs of employees must be able to communicate securely
- every pair shares a key – need $\approx \mathrm{U}^2$ keys altogether

  Q. How to distribute the keys? How to manage many secret keys?

Key Distribution Center (KDC). trusted entity, e.g. IT manager, as *intermediary*

- each employee shares a single key with KDC, generated on first day
- when Alice wants to communicate with Bob, KDC generates a *one-time session key*

  sends to Alice encrypted under Alice's key, and to Bob encrypted under Bob's key

Properties.

+ each employee stores *one* key, e.g. on smartcard
+ when employee X joins/leaves, all updates limited to X and KDC
− KDC single point of failure (c.f. security breaches or hardware failure)

Key Distribution Protocol

First Attempt.

1. Alice $\rightarrow$ KDC: "I wanted to talk to Bob"
2. KDC $\rightarrow$ Alice: $\mathrm{Enc}_{\mathrm{K}_{\mathrm{Alice}}}(\mathrm{k})$
3. KDC $\rightarrow$ Bob: "use this key to talk to Alice", $\mathrm{Enc}_{\mathrm{K}_{\mathrm{Bob}}}(\mathrm{k})$

- problem. KDC will "hang" indefinitely if Bob is not online

General template.

1. Alice $\rightarrow$ KDC: "I wanted to talk to Bob"
2. KDC $\rightarrow$ Alice: $\mathrm{Enc}_{\mathrm{K}_{\mathrm{Alice}}}(\mathrm{k})$, $\mathrm{Enc}_{\mathrm{K}_{\mathrm{Bob}}}(\mathrm{k})$
3. Alice $\rightarrow$ Bob: "Let's talk", $\mathrm{Enc}_{\mathrm{K}_{\mathrm{Bob}}}(\mathrm{k})$

- note. use encrypt then authenticate

Review: Arithmetic mod Composites

- Let $N = pq$ where p, q are prime
- Notation: $\mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\}$

  $\mathbb{Z}_N^*$ = invertible elements in $\mathbb{Z}_N$

  example: $N = 15 = 3 \cdot 5$, $\mathbb{Z}_N^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$

- Facts:

  $x \in \mathbb{Z}_N$ is in $\mathbb{Z}_N^* \iff \gcd(x, N) = 1$

  number of elements in $\mathbb{Z}_N^*$ is $\phi(N) = (p-1)(q-1)$

  Euler's theorem: $\forall\, x \in \mathbb{Z}_N^* : x^{\phi(N)} = 1$

  if $e \cdot d = 1 \pmod{\phi(N)}$, then $(x^e)^d = x \bmod N$

- Algorithms:

  Can add, mutiple, compute gcd, exponentiations and inverses mod N efficiently

Trapdoor Permutations

Three algorithms (G, F, F^{-1}):

- G outputs (pk, sk), pk defines a permutation $F(pk, \cdot) : X \to X$
- $F(pk, x)$ evaluates the function at x
- $F^{-1}(sk, y)$ inverts the function at y using sk (the trapdoor)

- security: given random pk, y, hard to compute pre-image of y

RSA trapdoor permutation (1977).

- G: outputs (N, e) as pk, where $N = pq$ approx 1024 bits, p, q approx 512 bits

  e encryption exponent $\gcd(e, \phi(N)) = 1$.
- F: $\text{RSA}(x) = x^e \bmod N$
- F^{-1}: trapdoor d such that $de = 1 \pmod{\phi N}$

  on input y, output $y^d \bmod N$