15-499:Algorithms and Applications

Cryptography I – Introduction
- Terminology
- Some primitives
- Some protocols

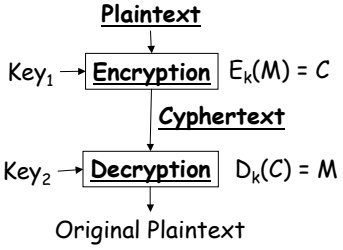
Cryptography Outline

Introduction: terminology and background
Primitives: one-way hash functions, trapdoors, …
Protocols: digital signatures, key exchange, ..
Number Theory: groups, fields, …
Private-Key Algorithms: DES, Rijndael, RC4
Cryptanalysis: Differential, Linear
Public-Key Algorithms: Knapsack, RSA, El-Gamal, Blum-Goldwasser
Case Studies: Kerberos, Digital Cash

Some Terminology

Cryptography – the general term
Cryptology – the mathematics
Encryption – encoding but sometimes used as general term)
Cryptanalysis – breaking codes
Steganography – hiding message
Cipher – a method or algorithm for encrypting or decrypting

More Definitions



Private Key or **Symmetric**: $Key_1 = Key_2$
Public Key or **Asymmetric**: $Key_1 \neq Key_2$
Key_1 or Key_2 is public depending on the protocol

Cryptanalytic Attacks

C = ciphertext messages
M = plaintext messages

Ciphertext Only: Attacker has multiple **C**s but does not know the corresponding **M**s

Known Plaintext: Attacker knows some number of **(C,M)** pairs.

Chosen Plaintext: Attacker gets to choose **M** and generate **C**.

Chosen Ciphertext: Attacker gets to choose **C** and generate **M**.

What does it mean to be secure?

Unconditionally Secure: Encrypted message cannot be decoded without the key

Shannon showed in 1943 that key must be as long as the message to be unconditionally secure – this is based on information theory

A **one time pad** – xor a random key with a message (Used in 2nd world war)

Security based on computational cost: it is computationally "infeasible" to decode a message without the key.

No (probabilistic) polynomial time algorithm can decode the message.

The Cast

Alice – initiates a message or protocol
Bob - second participant
Trent – trusted middleman
Eve – eavesdropper
Mallory – malicious active attacker

Primitives

One-way functions
One-way trapdoor functions
One-way hash functions
One-way permutations

Primitives: One-Way Functions

A function

y = f(x)

*is **one-way** if it is easy to compute **y** from **x** but "hard" to compute **x** from **y***

Building block of most cryptographic protocols

And, the security of most protocols rely on their existence.

Unfortunately, not known to exist. This is true even if we assume **P ≠ NP**.

One-way functions: possible definition

1. F(x) is polynomial time
2. $F^{-1}(x)$ is NP-hard

It is not hard to come up with such functions.

But, what is wrong with this?

One-way functions: better definition

For most **y** no single PPT (probabilistic polynomial time) algorithm can compute **x**

Roughly: at most a fraction $1/|x|^k$ instances **x** are easy for any **k** and as $|x| \to \infty$

This definition can be used to make the probability of hitting an easy instance arbitrarily small.

Some examples (conjectures)

Factoring:

x = (u,v)

y = f(u,v) = u*v

If u and v are prime it is hard to generate them from y.

Discrete Log: $y = g^x \bmod p$

where p is prime and g is a "generator" (*i.e.*, g^1, g^2, g^3, … generates all values < p).

DES with fixed message: $y = DES_x(m)$

This would assume a family of DES functions of increasing size

One-way functions in private-key protocols

y = ciphertext
m = plaintext
x = key
$y = f(x) = E_x(m)$
In a **known-plaintext attack** we know a (y,m) pair.
The m along with E defines f(x)
f(x) needs to be easy
$f^{-1}(y)$ should be hard
Otherwise we could extract the key x.

One-way functions in public-key protocols

y = ciphertext
x = plaintext
k = public key
$y = f(x) = E_k(x)$
We know k and thus f(x)
f(x) needs to be easy
$f^{-1}(y)$ should be hard
Otherwise we could decrypt y.
But what about the intended recipient, who should be able to decrypt y?

Note the change of role of the key and plaintext from the previous example

One-Way Trapdoor Functions

*A **one-way** function with a "trapdoor"*
*The **trapdoor** is a key that makes it easy to invert the function y = f(x)*
Example: **RSA** (conjecture)
$y = x^e \bmod n$
Where n = pq (p, q, e are prime)
p or q or d (where ed = (p-1)(q-1) mod n) can be used as trapdoors
In public-key algorithms
f(x) = public key (*e.g.,* e and n in RSA)
Trapdoor = private key (*e.g.,* d in RSA)

One-way Hash Functions

Y = h(x) where
- y is a fixed length independent of the size of x. In general this means h is not invertible since it is many to one.
- Calculating y from x is easy
- Calculating any x such that y = h(x) give y is hard

Used in digital signatures and some other protocols.

Protocols

Built out of the primitives.
Often the weakest point in terms of security

- Digital signatures
- Authentication
- Key exchange
- Secret sharing
- Timestamping services
- Zero-knowledge proofs
- Blind-signatures
- Key-escrow
- Secure elections
- Digital cash

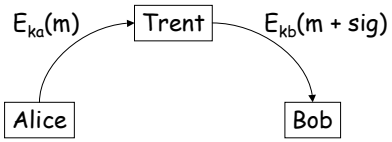
Protocols: Digital Signatures

Goals:

1. Convince recipient that message was actually sent by a trusted source
2. Do not allow repudiation, *i.e.,* that's not my signature.
3. Do not allow tampering with the message without invalidating the signature

Item 2 turns out to be really hard to do

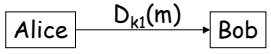
Using private keys



- ka is a secret key shared by Alice and Trent
- kb is a secret key shared by Bob and Trent

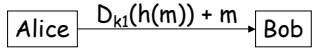
Sig is a note from Trent saying that Alice "signed" it.
To prevent repudation Trent needs to keep m or at least h(m) in a database

Using Public Keys



K1 = Alice's private key
Bob decrypts it with her public key

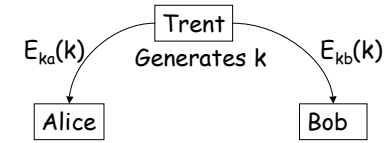
More Efficiently



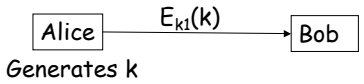
h(m) is a one-way hash of m

Key Exchange

Private Key method



Public Key method



Generates k

k1 = Bob's public key