Lecture S1: Cryptology

Cryptology

Cryptography: science of creating secret codes.
Cryptanalysis: science of code breaking.
Cryptology: science of secret communication.

Goal: information security in presence of malicious adversaries.

- Confidentiality.
- Integrity.
- Authentication.
- Authorization.
- Non-repudiation.

Analog Cryptography

Task.	Analog implementation.
Protect information.	
Identification.	
Contract.	
Money transfer.	
Public auction.	
Poker.	
Public election.	
Public lottery.	
Anonymous communication.	

Analog Cryptography

Task.	Analog implementation.
Protect information.	Code book, lock + key.
Identification.	Driver's license, SS, password.
Contract.	Handwritten signature, notary.
Money transfer.	Coin, bill, check, credit card.
Public auction.	Sealed envelope.
Poker.	Cards with concealed back.
Public election.	Anonymous ballot.
Public lottery.	Dice, coins.
Anonymous communication.	Pseudonym, "ransom note."

PaSS Me or Die!
- S. CLeMenS

Digital Cryptography

Our goal.

- Implement all tasks digitally and securely.
- Implement additional tasks that can't be done with physics!
  - play poker over phone
  - anonymous elections where everyone learns winner, but nothing else

Fundamental questions.

- Is any of this possible?
- How?

Today.

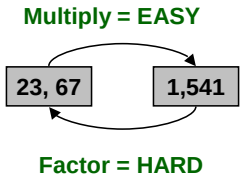
- Give flavor of modern (digital) cryptography.
- Implement a few of these tasks.
- Sketch a few technical details.

Digital Cryptography Axioms

Axiom 1. Players can toss coins.

Axiom 2. Players are computationally limited.

Axiom 3. Factoring is hard computationally.



Fact. Primality testing is easy.

Theorem. Digital cryptography exists.

Encryption

Encryption.

- Most basic problem in cryptography.
- Alice wants to send Bob a private message m.

Encryption

Encryption.

- Most basic problem in cryptography.
- Alice sends Bob an encrypted message E(m).
- Easy for Bob to recover original message m.
- Hard for Eve to learn anything about m.

Encryption

Alice sends Bob a message.

- Encode message as binary string (e.g., ASCII).
- Alice and Bob share secret key k.
- Everything else is public.

Private Key Encryption: One Time Pad

Key distribution.

- Alice and Bob share N-bit secret key k.

^ means bitwise XOR

Alice wants to send N-bit message m to Bob.

- Alice computes E(m) = m ^ k and sends E(m).

Bob receives ciphertext c = E(m).

- Bob compute D(c) = c ^ k.

0 1 0 1 1 0	m
0 1 1 1 0 0	k
0 0 1 0 1 0	m ^ k
0 1 0 1 1 0	(m ^ k) ^ k

Why does it work?

- D(E(m)) = D(m ^ k) = (m ^ k) ^ k = m

Why is it secure?

- If k is uniformly random, so is m ^ k.

Private Key Encryption

Advantages.

- Provably secure if key is random.
- Simple to implement.

Disadvantages.

- Hard to generate uniformly random keys.
- Need new key for each message.
- Signature?
- Non-repudiation?
- Key distribution?

A Russian one-time pad captured by MI5.

Other private key encryption schemes.

- Data Encryption Standard (DES).
- Advanced Encryption Standard (AES, Rijndael algorithm).
- Blowfish.

Public Key Encryption

Alice sends Bob a message.

- Bob has PUBLIC and PRIVATE keys.
  - public key locks, private key opens
- Everything else is public.

Public Key Encryption

Key distribution.

- Bob has PUBLIC key = published in digital phonebook (VeriSign).
- Bob has PRIVATE key = known only by Bob.

Alice wants to transmit N-bit private message m to Bob.

- Alice encrypts message using Bob's public key: E().

Bob receives ciphertext c from Alice.

- Bob decrypts message using his private key: D().

Under what situations does it work? D(E(m)) = m.

What are necessary conditions for security?

- Can encrypt message efficiently with public key.
- Can decrypt message efficiently with private key.
- CANNOT decrypt message efficiently with public key alone.

Modular Arithmetic

Do all computations modulo some base n.

- $10 + 4 \pmod{12} = 2$
- $38 * 15 \pmod{280} = 570 \pmod{280} = 10$

RSA Public-Key Cryptosystem

RSA cryptosystem (Rivest-Shamir-Adleman, 1978).

- Most widely used public-key cryptosystem (500 million users).
- Sun, Microsoft, Apple, browsers, cell phones, ATM machines, . . .
- Based on difference between testing primality and factoring.

Key generation.

- Select two large prime numbers p and q at random.
- Compute n = pq.

p = 11, q = 29
n = 319
e = 3, d = 187
m = 100

Fact. If p and q are prime, there exist efficiently computable integers e and d such that:

- For all m: $(m^e)^d \equiv m \pmod{n}$

Bob's keys.

- Public key: (e, n).
- Secret key: (d, n).

RSA Public-Key Cryptosystem

Alice wants to transmit N-bit private message m to Bob.

- Alice obtains Bob's public key (e, n) from Internet.
- Alice computes $E(m) = m^e \pmod{n}$.

m < n

Bob receives ciphertext c from Alice.

- Bob uses his secret key (d, n).
- Bob computes $D(c) = c^d \pmod{n}$.

Why does it work? Need to check that D(E(m)) = m.

- $D(E(m)) \equiv D(m^e) \pmod{n}$
  $\equiv (m^e)^d \pmod{n}$
  $\equiv m \pmod{n}$

RSA Example

Parameters.

- p = 47, q = 79, n = 3713, e = 17, d = 3377, m = 2003

Modular exponentiation.

- $2003^{17} \pmod{3713}$
  - $\equiv$ 1344547464276713705683401954485709119669029986291256541 63 $\pmod{3713}$
  - $\equiv$ 232

Fact. Can mod out by n after each multiplication.

- Intermediate numbers stay small.

Analysis.

- Suppose x, y, n are N-bit integers.
- # multiplications proportional to 2^N.

brute force

```
power(x, y, n) {
   c = 1
   repeat y times
      c = (c * x) % n
   return c
}
```


Repeated Squaring

Parameters.

- p = 47, q = 79, n = 3713, e = 17, d = 3377, m = 2003
- $m^e \bmod n = 232$

Efficient alternative. (repeated squaring) $17 = 10001_2$

- $2003^1 \pmod{3713} \equiv 2003 \pmod{3713} \equiv 2003$ 1
- $2003^2 \pmod{3713} \equiv 4{,}012{,}009 \pmod{3713} \equiv 1969$ 0
- $2003^4 \pmod{3713} \equiv 1969^2 \pmod{3713} \equiv 589$ 0
- $2003^8 \pmod{3713} \equiv 589^2 \pmod{3713} \equiv 1612$ 0
- $2003^{16} \pmod{3713} \equiv 1612^2 \pmod{3713} \equiv 3157$ 1

$$
\begin{aligned}
2003^{17} &\pmod{3713} \\
&\equiv 2003^{16} * 2003^{1} \pmod{3713} \\
&\equiv 3157 * 2003 \pmod{3713} \\
&\equiv 6323471 \pmod{3713} \\
&\equiv 232 \pmod{3713}
\end{aligned}
$$

Repeated Squaring

Repeated squaring.

$$
x^y = \begin{cases} 1 & \text{if } y = 0 \\ x^{y/2} \cdot x^{y/2} & \text{if } y \text{ is even} \\ x \cdot x^{(y-1)/2} \cdot x^{(y-1)/2} & \text{if } y \text{ is odd} \end{cases}
$$

Analysis.

- Suppose x, y, n are N-bit integers.
- All intermediate integers $\le$ 2N-bits.
- y decreases by a factor of 2 after at most 2 multiplications and mods.
- # multiplications (or mods) $\le$ 2N.

repeated squaring

```
power(x, y, n) {
   if (y == 0)
      return 1
   t = power(x, y/2)
   c = (t * t) % n
   if (y is even)
      return c
   else
      return (x * c) % n
}
```


RSA Details

How large should n = pq be?

- 1,024 bits for long term security.
- Too small $\Rightarrow$ easy to break.
- Too large $\Rightarrow$ time consuming to encrypt/decrypt.

How to choose large "random" prime numbers?

- Number theory $\Rightarrow$ n / $\log_e$ n prime numbers between 2 and n.
- Use Miller-Rabin algorithm to check whether integer is prime.

RSA Attacks

Factoring.

- Factor n = pq.
- d is public.
- Use p, q, and d to compute e.

Other means?

- Long-standing open research question.
- No guarantee that RSA is secure even if factoring is hard.

RSA Tradeoffs

Advantages.

- One public and one private key per INDIVIDUAL (not per message).
- Digital signatures.
- Relatively easy to implement.

Disadvantages.

- Security relies on decryption being "computationally inefficient."
- Relatively expensive to decrypt
  (often used in hybrid system, e.g., with DES).

RSA in the Real World

Secure Internet communication.

- Browsers.
- S/MIME, SSL, S/WAN.
- PGP.
- Microsoft Outlook.

Operating systems.

- Sun, Microsoft, Apple, Novell.

Hardware.

- Cell phones.
- ATM machines.
- Wireless Ethernet cards.
- Smart cards (Mondex).
- Palm Pilots.