Lecture 22: Cryptology

Overview

Turing machines. Newtonian mechanics.
Computability. Heisenberg uncertainty principle.
NP-completeness. Speed of light.

This lecture.

- Exploit hard problems.
- Apply theory to cryptography.
- RSA cryptosystem.

Cryptology

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

Goal: information security in presence of malicious adversaries.

- ⇨ Confidentiality: keep communication private.
- Integrity: detect unauthorized alteration to communication.
- Authentication: confirm identity of sender.
- Authorization: establish level of access for trusted parties.
- Non-repudiation: prove that communication was received.

A Better Approach

Security by obscurity.

- Rely on proprietary, ad hoc cryptographic schemes.
- Ex: CSS for DVD encryption, RIAA digital watermarking, GSM cell phones, Windows XP product activation, Adobe eBooks, ...
- Eventually reverse-engineered and cracked.

⇨ A better approach.

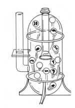
- Leverage theory of hard problems.
- Show that breaking security system is equivalent to solving some of the world's greatest unsolved problems!

Kerchoff's principle. *The system should not depend on secrecy, and it should be able to fall into the enemy's hands without disadvantage.*

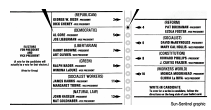
Analog Cryptography

Task	Description
Protect information	Code book, lock + key
Identification	Driver's license, fingerprint, DNA
Contract	Handwritten signature, notary
Money transfer	Coin, bill, check, credit card
Public auction	Sealed envelope
Poker	Cards with concealed backs
Public election	Anonymous ballot
Public lottery	Dice, coins
Anonymous communication	Pseudonum, ransom note









Digital Cryptography

Our goal.
- Implement all tasks digitally and securely.
- Implement additional tasks that can't be done with physics!

Fundamental questions.
- Is any of this possible?
- How?

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

Digital Cryptography Axioms

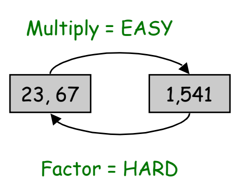
Axiom 1. Players can toss coins.
- Crypto impossible without randomness.

Axiom 2. Players are computationally limited.
- Polynomial time.

Axiom 3. Factoring is hard computationally.
- Not polynomial-time.
- "1-way trapdoor function."

Fact. Primality testing is easy computationally.

Theorem. Digital cryptography exists.
- Can do all previous tasks DIGITALLY.



Non-Encryption

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

credit card number

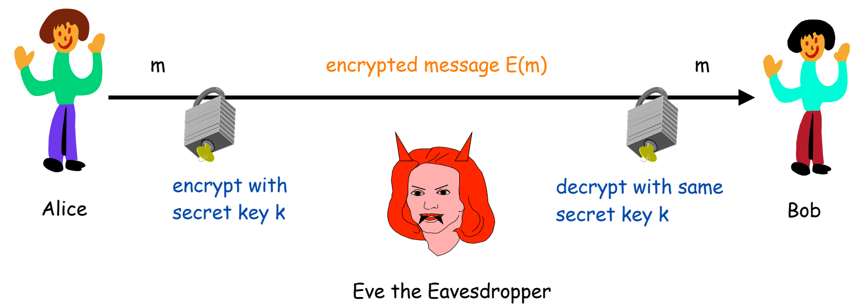
Encryption

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.

credit card number

Private Key Encryption

Alice sends Bob a message m.
- Assume message m encoded in binary.
- Alice and Bob share secret key k.



Private Key Encryption: One Time Pad

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

Alice wants to send N-bit message m to Bob.
- Alice computes and sends E(m) = m ^ k.

bitwise XOR

Bob receives ciphertext c = E(m).
- Bob computes D(c) = c ^ k.

0	1	1	1	0	0	k
N = 6						
0	1	0	1	1	0	m
0	0	1	0	1	0	E(m)
0	0	1	0	1	0	c
0	1	0	1	1	0	D(c)

Why does it work? D(E(m)) = D(m ^ k) = (m ^ k) ^ k = m

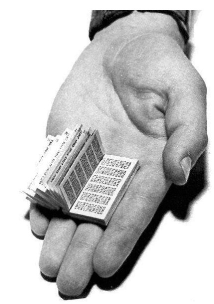
Why is it secure? If k is uniformly random, so is m ^ k.

Private Key Encryption

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

Disadvantages.
- Not easy to generate uniformly random keys.
- Need new key for each message.
- Signature?
- Non-repudiation?
- Key distribution? deal-breaker for e-commerce since Alice and Bob want to communicate even if they've never met

Other private key encryption schemes.
- Data Encryption Standard (DES).
- Advanced Encryption Standard (AES, Rijndael algorithm).
- Blowfish.



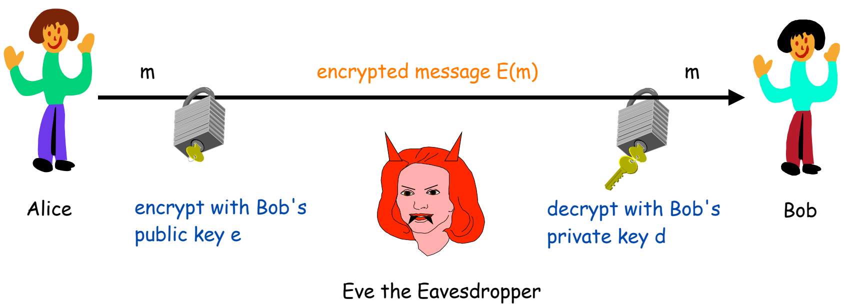
A Russian one-time pad.

Public Key Encryption

Alice sends Bob a message m.

- Bob has PUBLIC key e and PRIVATE key d.

locks unlocks



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(m).

Bob receives ciphertext c = E(m) from Alice.

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

Under what situations does it work? D(E(m)) = m. absolute and obvious requirement

What are necessary conditions for security?

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

RSA Public Key Cryptosystem: In the Real World

RSA cryptosystem (1978).

Rivest Shamir Adleman

Secure Internet communication. Browsers, S/MIME, SSL, S/WAN, PGP, Microsoft Outlook, etc.

> Alice browses to `https://whiteboard.cs.princeton.edu`
> Alice's browser gets Bob's public key.
> Alice sends programming assignment.
> Bob's web server decrypts assignment.

Alice submits programming assignment to Bob via secure website

Operating systems. Sun, Microsoft, Apple, Novell.

Hardware. Cell phones, ATM machines, wireless Ethernet cards, Mondex smart cards, Palm Pilots, Palladium.

RSA Public-Key Cryptosystem: Key Generation

RSA key generation.

- Select two large prime numbers p and q at random. $p = 11, q = 29$
- Compute n = pq. $n = 11 \times 29 = 319$

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

- For all messages m: $(m^e)^d \equiv m \pmod n$ $\quad (m^3)^{187} \equiv m \pmod{319}$

$a \equiv b \pmod n$ means (a % n) == (b % n)

Bob's public key: (e, n) (3, 319)

Bob private key: (d, n) (187, 319)

RSA Public-Key Cryptosystem: Encryption and Decryption

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

m = 100

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

$E(m) = 100^3 \pmod{319}$
$= 254$

Bob receives ciphertext c from Alice.

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

$D(c) = 254^{187} \pmod{319}$
$= 100$

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

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

previous fact

Modular Exponentiation: Brute Force

Modular exponentiation: $c = a^b \pmod n$.

$2003^{17} \pmod{3713} = 134454746427671370568340195448570911966902998629125654163 \pmod{3713}$
$= 232$

Brute force: multiply a by itself, b times.

Analysis of brute force.

- Suppose a, b, and n are N-bit integers.
- Problem 1: number of multiplications proportional to 2^N.
- Problem 2: number of digits of intermediate value can be 2^N.
- Exponential time and memory!

very bad news since N must be big for RSA to be secure

128TB memory if N = 50

Modular Exponentiation: Repeated Squaring

Idea 1. Can mod out by n after each multiplication.

- Intermediate numbers stay small.

Idea 2. Repeated squaring.

$$\begin{aligned} &2003^{17} &\pmod{3713} \\ &\equiv 2003^1 \times 2003^{16} &\pmod{3713} \\ &\equiv 2003 \times 3157 &\pmod{3713} \\ &\equiv 6{,}323{,}471 &\pmod{3713} \\ &\equiv 232 &\pmod{3713} \end{aligned}$$

$17_{10} = 10001_2$

Term	To compute	mod 3713
2003^1	2003	2003
2003^2	2003^2	1969
2003^4	1969^2	589
2003^8	589^2	1612
2003^{16}	1612^2	3157

Repeated squaring

Analysis of modular exponentiation.

- At most 2N multiply and mod operations.
- Intermediate numbers at most 2N digits long.

RSA Details

How large should n = pq be?

- 2,048 bits for long term security.
- Too small ⇒ easy to break.
- Too large ⇒ time consuming to encrypt/decrypt.

How to choose large "random" prime numbers?

Prime Number Theorem. (Hadamard, Vallée Poussin, 1896).
Asymptotically, there are $n / \log_e n$ prime numbers between 2 and n.

- Primes are plentiful: 10^{151} with ≤ 512 bits.
- Will never run out, and no two people will pick same ones.

Theorem (Agarwal-Kayal-Saxena, 2002). PRIME is in P.

- PRIME: Given integer n, is n prime?

RSA in Java

Key generation using:

- java.math.BigInteger, java.security.SecureRandom

```
SecureRandom random = new SecureRandom();

BigInteger ONE = new BigInteger("1");   random N/2-bit prime
BigInteger p   = BigInteger.probablePrime(N/2, random);
BigInteger q   = BigInteger.probablePrime(N/2, random);
BigInteger phi = (p.subtract(ONE)).multiply(q.subtract(ONE));

BigInteger n = p.multiply(q);                    modulus
BigInteger e = new BigInteger("65537");          public key
BigInteger d = e.modInverse(phi);                private key
```

(ed ≡ 1 mod ϕ)

RSA function.

```
BigInteger rsa(BigInteger a, BigInteger b, BigInteger n) {
   return a.modPow(b, n);
}        built-in modular exponentiation (repeated squaring)
```


Cryptanalysis: RSA Attacks

Factoring. Factor n = pq. Use p, q, and e to compute d.

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

Semantic security. If you know Alice will send ATTACK or RETREAT you can encrypt ATTACK and RETREAT using Bob's public key, and check which one Alice sent.

Timing attack. Alice gleans information about Bob's private key by measuring time it takes Bob to exponentiate.

Modulus sharing.

- Bob: (d_1, e_1, n), Ben: (d_2, e_2, n).
- Bob can compute d_2 given e_2; Ben can compute d_1 given e_1.

RSA Tradeoffs

Advantages.

- Solves key distribution problem.
- Extends to digital signatures, etc.

Disadvantages.

no such reliance with one-time pads

- Security relies on decryption being "computationally inefficient."
- Not semantically secure.
- Decryption more expensive than private key schemes.

Practical middle-ground hybrid system.

- Use AES, a private key encryption system.
- Use RSA to distribute AES keys.

Theoretical high-ground. Blum-Goldwasser (1985)

- Provably as hard a factoring.
- Semantically secure.

Consequences of Cryptography

Crypto liberates. you = Alice or Bob

- Freedom of privacy, speech, press, political association.
- Benefits both ordinary citizens and terrorists.

Crypto enables e-commerce. confidentiality, integrity, authentication.

> *Encrypting transactions on the Internet is the equivalent of arranging an armored car to deliver credit-card information from someone living in a cardboard box to someone living on a park bench.*
>
> - Eugene Spafford

Crypto restricts. you = Eve, your computer = Alice or Bob

- Ex: Trustworthy Computing.
- Establishes a secure identity and enable secure transactions.
- Restricts what user can do (play MP3 files, copy DVDs, run software, print documents, forward email).
- Brought to you by Microsoft in 2005.