7.9 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.

Introduction to Computer Science • Robert Sedgewick and Kevin Wayne • Copyright © 2005 • http://www.cs.princeton.edu/introcs

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.

A Better Approach

Security by obscurity.
- Rely on proprietary, ad hoc cryptographic schemes.
- 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!

Kerckhoffs’ principle.
- The system must not require secrecy and can be stolen by the enemy without causing trouble.

“It is insufficient to protect ourselves with laws. We need to protect ourselves with mathematics.”
- Bruce Schneier

“Cryptography used to be an obscure science, of little relevance to everyday life. Historically, it always had a special role in military and diplomatic communications. But in the Information Age, cryptography is about political power, and in particular, about the power relationship between a government and its people. It is about the right to privacy, freedom of speech, freedom of political association, freedom of the press, freedom from unreasonable search and seizure, freedom to be left alone.”
- Phil Zimmermann
Analog Cryptography

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

Digital Cryptography Axioms

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

**Axiom 2.** Players are computationally limited (poly-time).

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

**Fact.** Primality testing is easy computationally.

**Theorem.** Digital cryptography exists.

**Corollary.** Can do all tasks on previous slide digitally.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiply</td>
<td>EASY</td>
</tr>
<tr>
<td>Factor</td>
<td>HARD</td>
</tr>
</tbody>
</table>

Non-Encryption

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

Alice

Eve the Eavesdropper

Bob

Message m

credit card number

Multiply = EASY

Factor = HARD

23, 67 1,541
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 $m$.

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 $E(m) = m \oplus k$.
- Bob receives ciphertext $c = E(m)$.
  - Bob computes $D(c) = c \oplus k$.

Why does it work? $D(E(m)) = D(m \oplus k) = (m \oplus k) \oplus k = m$.
Why is it secure? If $k$ is uniformly random, so is $m \oplus k$.

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?

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

Russian one-time pad

Deal-breaker for e-commerce since Alice and Bob want to communicate even if they've never met.
Public Key Encryption

**RSA Public Key Cryptosystem: In the Real World**

**RSA cryptosystem.** [Rivest-Shamir-Adleman 1978]

**Operating systems.** Sun, Microsoft, Apple, Novell.

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

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

Alice sends Bob a message m.

- Bob has **public key** e and **private key** d.

Alice encrypts message using Bob’s **public key**: $E(m)$.

Bob decrypts message using his **private key**: $D(C) = m$.

Bob has **public key** = published in digital phonebook.

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: Key Generation**

**RSA key generation.**

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

**Number theory fact.** If $p$ and $q$ are prime, there exist efficiently computable integers $e$ and $d$ such that for all messages $m$: $(m^e)^d = m \pmod{N}$.

- $a = b \pmod{N}$ means $(a \% N) = (b \% N)$

Bob’s **public key**: $(e, N)$

Bob’s **private key**: $(d, N)$

---

Alice sends Bob a message $m$.

- Bob has **public key** $e$ and **private key** $d$.

Alice encrypts message using Bob’s **public key**: $E(m)$.

Bob decrypts message using his **private key**: $D(C) = m$.

Bob has **public key** = published in digital phonebook.

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.** [Rivest-Shamir-Adleman 1978]

**Operating systems.** Sun, Microsoft, Apple, Novell.

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

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

Alice sends Bob a message $m$.

- Bob has **public key** $e$ and **private key** $d$.

Alice encrypts message using Bob’s **public key**: $E(m)$.

Bob decrypts message using his **private key**: $D(C) = m$.

Bob has **public key** = published in digital phonebook.

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: Key Generation**

**RSA key generation.**

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

**Number theory fact.** If $p$ and $q$ are prime, there exist efficiently computable integers $e$ and $d$ such that for all messages $m$: $(m^e)^d = m \pmod{N}$.

- $a = b \pmod{N}$ means $(a \% N) = (b \% N)$

Bob’s **public key**: $(e, N)$

Bob’s **private key**: $(d, N)$
RSA Public-Key Cryptosystem: Encryption and Decryption

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 (mod N).

Bob receives ciphertext c from Alice.
- Bob uses his secret key (d, N).
- Bob computes D(c) = c^d (mod N).

Why does it work?
- Need to check that D(E(m)) = m.
- D(E(m)) = D(m^e) (mod N)
  = (m^e)^(d^-1) (mod N)
  = m^f (mod N)
  = m (mod N)

Modular Exponentiation: Repeated Squaring

Idea 1: can mod out by N after each multiplication.
- Intermediate numbers stay small.

Idea 2: 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.

Q. How do I choose a large “random” prime number?
A. Guess-and-check.

Prime Number Theorem. [Hadamard, Vallée Poussin, 1896]
- Number of primes between 2 and N ~ N / ln(N).
- Primes are plentiful: 10^{50} with ≤ 512 bits.
- Will never run out, and no two people will pick same ones.

Theorem. [Agarwal-Kayal-Saxena, 2002]
- PRIME: Given n-bit integer N, is N prime?
- PRIME is in P.
**RSA Tradeoffs**

**Advantages.**
- Solves key distribution problem.
- Extends to digital signatures, etc.
- No such reliance with one-time pads.

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

**Theoretical high-ground.** [Blum-Goldwasser, 1985]
- Provably as hard a factoring.
- Semantically secure.

**Practical middle-ground hybrid system.**
- Use AES, a fast private key encryption system.
- Use RSA to distribute AES keys.

---

**RSA in Java**

```java
SecureRandom random = new SecureRandom();
BigInteger ONE = new BigInteger("1");
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);
BigInteger e = new BigInteger("65537");
BigInteger d = e.modInverse(phi);
```

```
BigI nteger rsa(BigInteger a, BigInteger b, BigInteger N) {
    return a.modPow(b, N);
}
```

**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 gleams information about Bob’s private key by measuring time it takes Bob to exponentiate.

**Modulus sharing.**
- Bob: (d1, e1, N), Ben: (d2, e2, N).
- Bob can compute d2 given e2; Ben can compute d1 given e1.

**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, computer = Alice, speakers = Bob]
- Ex: Digital rights management (DRM).
- Establishes a secure identity and enable secure transactions.
- Restricts what user can do: play MP3 files, copy DVDs, run software, print documents, forward email.