# 7.9 Cryptology

"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



## 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.

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



## 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.
- Eventually reverse-engineered and cracked.
- Ex: CSS for DVD encryption, RIAA digital watermarking, GSM cell phones, Windows XP product activation, Adobe eBooks, Diebold AccuVote-TS machines, . . . .

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!

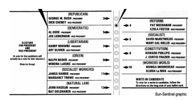
Kerckhoffs' principle.

"The system must not require secrecy and can be stolen by the enemy without causing trouble."

## Analog Cryptography

| Task | Description |
|---|---|
| Protect information | Code book, lock + key |
| 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 |
| Identification | Driver's license, fingerprint, DNA |
| Anonymous communication | Pseudonym, 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.

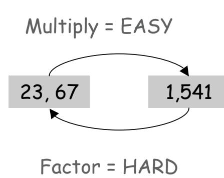


## Digital Cryptography Axioms

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

Axiom 2. Players are computationally limited (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.



## Non-Encryption

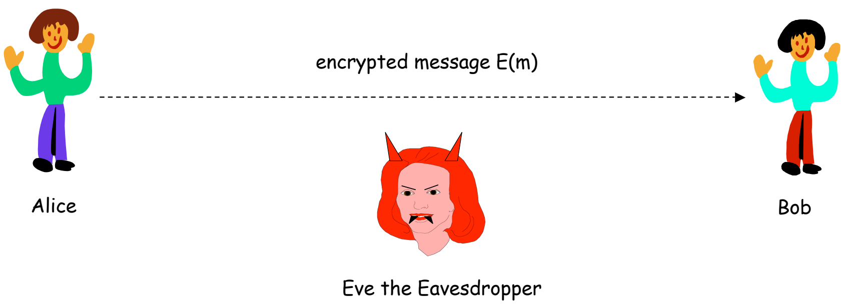
Encryption.
- Most basic problem in cryptography.
- Alice wants to send Bob a private message m.

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.

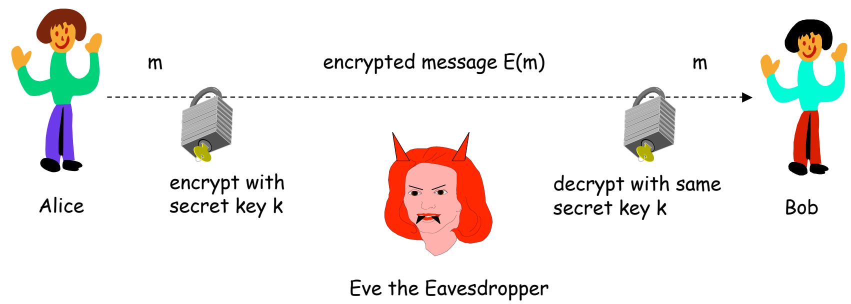
credit card number






## 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.

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 | k |

n = 6

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

bitwise XOR

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | 0 | m |
| 0 | 0 | 1 | 0 | 1 | 0 | E(m) |

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

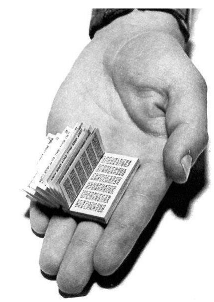
| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 0 | c |
| 0 | 1 | 0 | 1 | 1 | 0 | D(c) |

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



Russian one-time pad

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?

Rosenbergs sent to electric chair because Russian spy reused a one-time pad

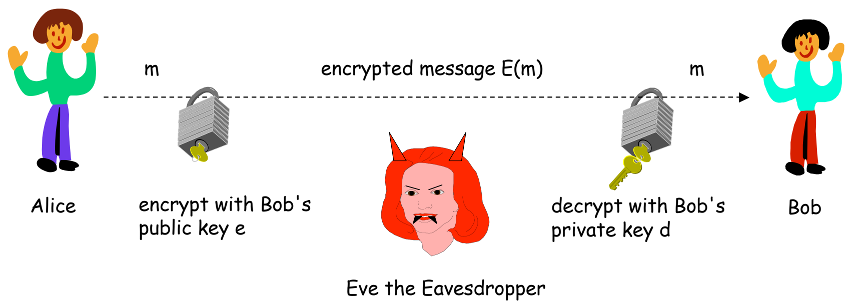
deal-breaker for e-commerce since Alice and Bob want to communicate even if they've never met

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

## Public Key Encryption

Alice sends Bob a message m.

- Bob has public key e and private key d.
  - e: locks
  - d: 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. [Rivest-Shamir-Adleman 1978]

2002 Turing award

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 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



## 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 × 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: $(m^e)^d \equiv m \pmod N$. — $(m^3)^{187} \equiv m \pmod{319}$

$a \equiv b \pmod N$ 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.

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

Bob receives ciphertext c from Alice.

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

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

m = 100

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

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: $c = a^b \pmod N$.

$$\begin{aligned} 2003^{17} \pmod{3713} &\equiv 134454746427671370568340195448570911966902998629125654163 \pmod{3713} \\ &\equiv 232 \end{aligned}$$

Brute force. Multiply a by itself, b times. Divide by N and keep remainder.

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!

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 | 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 $\Rightarrow$ easy to break.
- Too large $\Rightarrow$ 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 $\approx$ N / ln N.
- Primes are plentiful: $10^{151}$ with $\le$ 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 in Java

Key generation using: `java.math.BigInteger`, `java.security.SecureRandom`.

```
SecureRandom random = new SecureRandom();
                                           random n/2-bit prime
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);               modulus
BigInteger e = new BigInteger("65537");     public key
BigInteger d = e.modInverse(phi);           private key
```

(ed ≡ 1 mod $\phi$)

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.
- Security relies on decryption being "computationally inefficient." (no such reliance with one-time pads)
- 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.



## 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.