Lecture S1: Cryptology

Cryptology.
- Science of secret communication.

Goal: information security in presence of malicious adversaries.
- Confidentiality.
- Integrity.
- Authentication.
- Authorization.
- Non-repudiation.

Analog Cryptology

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

Analog implementation.

Digital Cryptology

Our goal.
- Implement all tasks digitally.
- 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 cryptology.
- Implemented a few of these tasks.
- Sketch a few technical details.
Digital Cryptology Axioms

Axiom 1.
- Players can toss coins.

Axiom 2.
- Players are computationally limited.

Axiom 3.
- Factoring is hard computationally.

Theorem.
- Digital cryptography exists.

Private Key Encryption

Assume message is encoded as binary string.
- ASCII.

Protocol.

Bob sends Alice message
M = original message
C = encrypted message

Bob and Alice share SECRET key. Everything else is public.

Public Key Encryption

Bob has N-bit message M to send Alice.
- Alice and Bob share N-bit private key K.
- Bob computes C = M ^ K and sends C.
- Alice receives C and computes C ^ K = (M ^ K) ^ K = M.

Advantage.

Disadvantage.

Two different keys:
Alice’s PUBLIC key locks, her PRIVATE key opens. Everything else is public.
Public Key Encryption

Bob has N-bit message to send to Alice.
- Alice has public and secret key.
  - PUBLIC key = published on Web in digital phonebook (VeriSign)
  - PRIVATE key = known only by Alice
- Bob encrypts message using Alice’s public key.
- Alice decrypts message using her private key.

To achieve security, need following properties:
- Can encrypt message efficiently with public key.
- Can decrypt message efficiently with private key.
- CANNOT decrypt message efficiently with public key alone.

RSA Public-Key Cryptosystem

- Most widely used public-key cryptosystem (500 million users).
- Sun, Microsoft, Apple, browsers, cell phones, ATM machines, . . .

Key generation.
- Select two large prime numbers p and q at random.
- Compute n = pq, and \( \phi = (p-1)(q-1) \).
- Choose integer e that is relatively prime to \( \phi \).
- Compute d such that \( de \equiv 1 \pmod{\phi} \).
- Publish (e, n) as public key.
- Keep (d, n) as secret key.

Note: don’t even need to keep p, q, or \( \phi \).
- \( \phi \) only needed to compute d.
- Saving p, q speeds up decryption (Chinese Remainder Theorem).

Modular Arithmetic

Do all computations modulo some base n.
- \( 10 + 4 \pmod{12} = 2 \)
- \( 38 \times 15 \pmod{280} = 570 \pmod{280} = 10 \)

RSA Public-Key Cryptosystem

Bob sends message M to Alice.
- Bob obtains Alice’s public key (e, n) from Internet.
- Bob computes \( C = M^e \pmod{n} \).

Alice receives message C.
- Alice uses her secret key (d, n).
- Alice computes \( M' = C^d \pmod{n} \).

Why does it work? Need \( M = M' \). Intuitively.
- \( M' = C^d \pmod{n} \)
  \( = M^{ed} \pmod{n} \)
  \( = M \) Recall: \( e d \equiv 1 \pmod{\phi} \).
- Argument not rigorous because of mod.
  - rigorous argument uses fact that p and q are prime, and
    \( \phi = (p-1)(q-1) \).
RSA Example

Parameters.

- \( p = 47, q = 79, n = 3713, \phi = 3588 \)
- \( e = 17, d = 3377 \)
- \( M = 2003 \)

Modular exponentiation.

- \( 2003^{17} \mod 3713 = 232 \)
- \( 2003^{1} \mod 3713 = 2003 \)
- \( 2003^{2} \mod 3713 = 4012,009 \mod 3713 = 1969 \)
- \( 2003^{4} \mod 3713 = 1969^{2} \mod 3713 = 589 \)
- \( 2003^{6} \mod 3713 = 589^{2} \mod 3713 = 1612 \)
- \( 2003^{16} \mod 3713 = 3157 \)

RSA Details

How large should \( n = pq \) be?

- 1,024 bits for long term security.
- IE, Netscape: 40, 56, 128 bit.
- Too small \( \Rightarrow \) easy to break.
- Too large \( \Rightarrow \) time consuming to encrypt/decrypt.

How to choose large “random” prime numbers?

- Miller-Rabin procedure checks whether \( x \) is prime. Usually!
- Number theory \( \Rightarrow n / \log_{2} n \) prime numbers between 2 and \( n \).

How to compute \( d \) efficiently?

- Existence guaranteed since \( \gcd(e, \phi) = 1 \).
- Fancy version of Euclid’s algorithm.

RSA Attacks

Factoring.

- Factor \( n = pq \).
- Then compute \( \phi \).
- Then compute \( e \).

Timing attacks.

- Reconstruct \( d \) by sending \( C \) and monitoring how long it takes to compute \( C^{d} \mod n \).

Other means?

- Long-standing open research question.

Note: Diffie-Helman cryptosystem can be broken if and only if factoring is hard.

- Discrete log: given \( x, n, C \), find \( d \) such that \( x^{d} \mod n = C \).

Digital Signature

Alice sends Bob a response.

- Bob’s wants to be sure Alice really sent it, and not some imposter.
RSA Digital Signature

Alice wants to send Bob a response $S$.
- Alice uses private key $d$ and computes: $S' = S^d \pmod{n}$.
- Alice sends $(S, S')$.

Bob receives digital signed response $(S, S')$.
- Bob uses Alice’s public key $e$ and checks if $S = (S')^e \pmod{n}$.
- If yes, then Bob concludes $S$ sent by Alice.
- If no, then Bob concludes $S$ or $S'$ corrupted in transmission, or message is a forgery.

Third party.
- Bob verifies Alice’s signature on digitally signed message (e.g., electronic check).
- Bob forwards digitally signed message to bank.
- Bank re-verifies Alice’s signature.

Note: $S^{ed} = S^{de} = S$ (commutativity)

RSA Tradeoffs

Advantages.

Disadvantages.

RSA Applications

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.

Bad Cryptology

Content Scrambling System (CSS).
- Used to encrypt DVD’s.
- Each disc has 3 40-bit keys.
- Each DVD decoder (software/hardware) has unique 40-bit key.
- "Not possible" to play back on computer without disc.

DeCSS. (Canman and SøupaFrog, 1999).
- Decryption algorithm written by two Norwegians
- Used "in-circuit emulator" to monitor hardware activity.

Why CSS is fatally flawed.
Why does it work? Rigorously.
- \( M^e \equiv C^d \pmod{n} \)
  \[ = M^{ed} \pmod{n} \]

Now, since \( \phi = (p-1)(q-1) \) and \( e \cdot d \equiv 1 \pmod{\phi} \)
- \( ed = 1 + k(p-1)(q-1) \) for some integer \( k \).

A little manipulation.
- \( M^{ed} = M \cdot M^{(p-1)(q-1)} \pmod{p} \)
  \[ = M \cdot 1^{k(q-1)} \pmod{p} \]
  \[ = M \pmod{p} \]
  (trivially true if \( M \equiv 0 \))
- \( M^{ed} = M \pmod{q} \)

Finally.
- \( M^{ed} = M \pmod{pq} \)