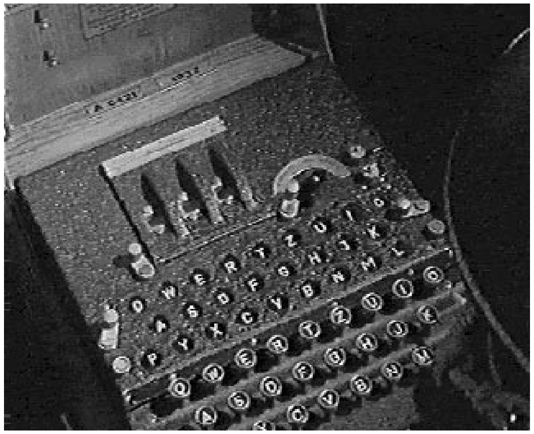
Lecture S1: Cryptology



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.

Implementation.

Digital Cryptologoy

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.

Multiply = EASY

23, 67 → 1,541

Factor = HARD

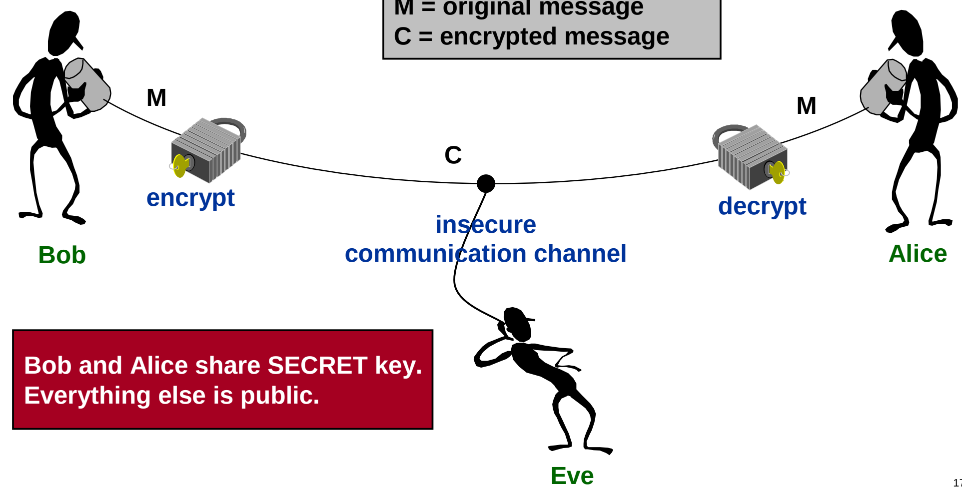
Theorem.
- Digital cryptography exists.

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.

Private 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.

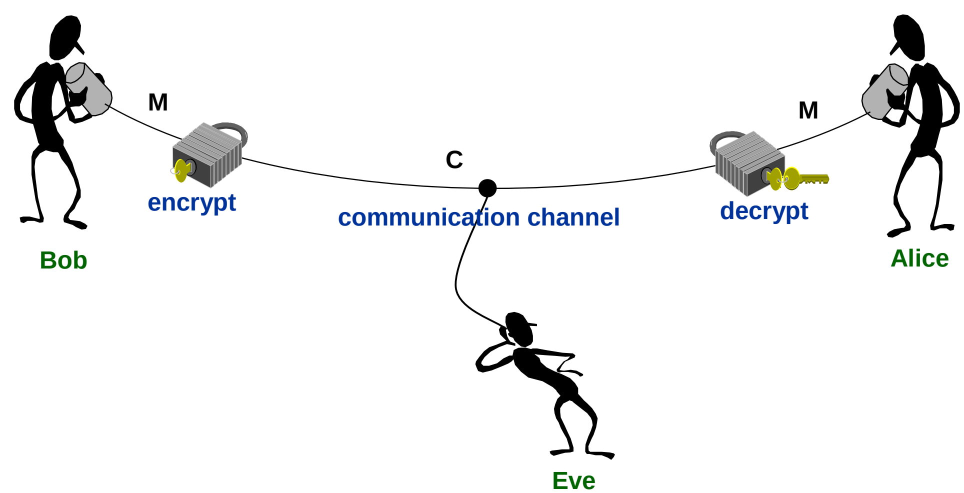
^ means bitwise XOR

Advantage.

Disadvantage.

0	1	0	1	1	0	M
0	1	1	1	0	0	K
0	0	1	0	1	0	C = M ^ K
0	1	0	1	1	0	(M ^ K) ^ K

Public Key Encryption



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.

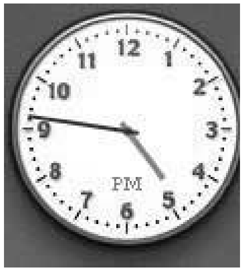
To achieve security, need following properties:

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

Modular Arithmetic

Do all computations modulo some base n.

- 10 + 4 (mod 12) = 2
- 38 * 15 (mod 280) = 570 (mod 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, . . .

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 ϕ.
- Compute d such that $d\,e \equiv e\,d \equiv 1 \pmod{\phi}$.
- Publish (e, n) as public key.
- Keep (d, n) as secret key.

p = 11, q = 29
n = 319, ϕ = 280
e = 3, d = 187
M = 100

Note: don't even need to keep p, q, or ϕ.

- ϕ only needed to compute d.
- Saving p, q speeds up decryption (Chinese Remainder Theorem).

RSA Public-Key Cryptosystem

Bob sends message M to Alice.

$\boxed{M < n}$

- 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.

- $$\begin{aligned} M' &\equiv C^d && \pmod{n} \\ &\equiv M^{ed} && \pmod{n} \\ &\equiv M && \text{Recall: } e\,d \equiv 1 \pmod{\phi}. \end{aligned}$$

- 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.

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

$$\begin{array}{ll} 2003^{17} & \pmod{3713} \\ = 2003^{16} * 2003^{1} & \pmod{3713} \\ = 3157 * 2003 & \pmod{3713} \\ = 6323471 & \pmod{3713} \\ = 232 & \end{array}$$

Modular exponentiation.

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

Better alternative (repeated squaring).

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

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 $\Rightarrow$ $n / \log_e n$ 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 ϕ.
- Then compute e.

Timing attacks.

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

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 $x^d \bmod n = C$.

RSA Digital Signature

Alice wants to send Bob a response S.

- Alice uses private key d and computes: $S' \equiv S^d \pmod{n}$.
- Alice sends (S, S').

Bob receives digital signed response (S, S').

- Bob uses Alice's public key e and checks if $S \equiv (S')^e \pmod{n}$.
- 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.

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

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.

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 Cryptologoy

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 SøupaFrøg, 1999).
- Decryption algorithm written by two Norwegians
- Used "in-circuit emulator" to monitor hardware activity.

Why CSS is fatally flawed.

Cryptography: Extra Slides

RSA Public-Key Cryptosystem

Why does it work? Rigorously.

- $M' = C^d \pmod{n}$
  $= M^{ed} \pmod{n}$

Now, since $\phi = (p-1)(q-1)$ and $e\,d \equiv 1 \pmod{\phi}$

- $ed = 1 + k(p-1)(q-1)$ for some integer k.

A little manipulation.

- $M^{ed} \equiv M\, M^{(p-1)\,k(q-1)} \pmod{p}$
  $\equiv M\,(1)^{k(q-1)} \pmod{p}$
  $\equiv M \pmod{p}$
  (trivially true if $M \equiv 0$)
- $M^{ed} \equiv M \pmod{q}$

Fermat's Little Theorem

if p is prime, then for all $a \neq 0$
$a^{p-1} \equiv 1 \pmod{p}$

Chinese Remainder Theorem

if p, q prime then for all x, a
$x \equiv a \pmod{pq} \Leftrightarrow$
$x \equiv a \pmod{p}$, $x \equiv a \pmod{q}$

Finally.

- $M^{ed} \equiv M \pmod{\underbrace{pq}_{n}}$