DSZQUPHSBQIZ

Cryptography

Intro

Why do we care about cryptography?

- Need for private data and private communication

    1. inherent to the human nature - kids are using crypto encoding and decoding to hide from parents, teachers, siblings, etc.
    2. essential in military conflicts
    3. essential in competitive situations - business, etc.

- The world is more and more interconnected $\Rightarrow$ need for crypto has increased.
- Importance in Computer Science: Cryptography provides the main tools for network security.

Examples:

(a) data needing security

- financial records
- medical records
- commercial secrets
- technical specifications
- business and private communications

(b) applications needing security

- electronic commerce
- electronic fund transfer (intra- and inter bank, ...)
- home banking, electronic cash, electronic data interchange

Who are the potential attackers?

- hackers
- industrial competitors
- spies
- press
- government agencies

Major players

NSA - National Security Agency

- created in 1952 by Pres. Truman
- Goals:
  - designing strong ciphers (to protect US communications)
  - breaking ciphers (to listen to non-US communications)
- very secretive: budget is not public
- largest employer of mathematicians in the world
- largest purchaser of computer hardware

RSA Security Inc.

- has patents for RSA, RC5, RC6, etc.
- over 500 mln. users of the basic crypto library BSAFE
- RSA Laboratory - research
- RSA Conference
- spin-off companies: VeriSign for Public Key Infrastructure

Many companies have introduced crypto into their products/services

Software:
Microsoft Lotus, Netscape, Oracle, Novell
Hardware:
IBM, Motorola, Intel, Sun, Hewlett-Packard
telecom
AT&T, Northern Telecom
finance
Visa, Mastercard, Verifone

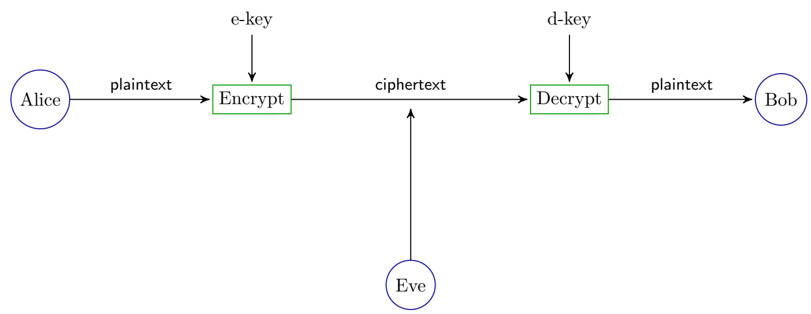
Standards in Cryptography
at different levels

- informal industrial standards:
  ex: RSA Labs: PKCS
- industrial standards
  ex: IEEE P1363
- banking standards
  ex: ANSI X.9
- federal standards, international standards
  NIST, FIPS ISO

Terminology

cryptology: all-inclusive term
cryptography: designing systems for enciphering data
cryptanalysis: breaking such systems

Basic Setting:



Eve's Goals

- get the message, or just some information about the message
- find the key and thus read all further messages
- replace Alice's message with another one, fooling Bob
- Masquerade as Alice and send Bob a message as if it comes from Alice

Types of attacks - depending on what info. Eve has

1. ciphertext only
2. known plaintext attack - Eve has one or more copies (plaintext, ciphertext)
3. chosen plaintext attack - Eve sees the ciphertext corresponding to a plaintext that she has chosen
4. chosen ciphertext attack - Eve sees the plaintext corresponding to a ciphertext that she has chosen

How do we get secrecy?

In the early days:
encryption method was kept secret (and security was based on this)

Herckhoff Principle 1883: we should assume that the enemy knows the encryption method.

So secrecy is based on the **key**. Consequence: the key has to be randomly chosen from a large set.

Two basic approaches:

1. symmetric cryptography
2. public key cryptography

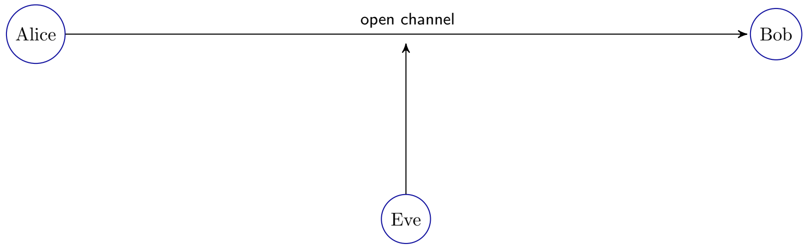
Symmetric Crypto

Alice and Bob share the e-key and the d-key (they can be the same or one can be easily derived from the other one)

1. Classical cryptosystems
2. DES, and its variants
3. AES
4. IDEA, Blowfish, Twofish, ...

Public-key Crypto

Suppose Alice and Bob are far apart from the onset.



Solution: public-key cryptography (PKC)

Bob has 2 keys: an encryption key and a decryption key

Bob makes his encryption key public, and keeps the decryption key secret

Necessary crucial property: the d-key cannot be determined from the e-key in a practical way.

RSA - most popular public key cryptosystem

El Gamal, NTRU, McEliece, knapsack, ...

One way to understand PKC - Physical metaphor

Bob places a box with a small opening in a public place

Alice puts her message in the box through the opening

Bob opens the box with a key that only he has

This can be implemented mathematically

There is a functional dependency between the e-key and the d-key but the d-key cannot be derived from the e-key in a reasonable amount of time

PKC depends on the existence of one-way functions. These are functions easy to compute in one direction and very hard to compute in the opposite direction.

No function has been proven to be one-way. But there is a strong belief that some functions are one-way.

Factoring large integers and the discrete log are functions believed to be one-way which are used in crypto.

Symmetric crypto is still used. Why: it is faster.

PKC is used for small amount of data, e.g., to exchange keys for symm. crypto.

Crypto Applications

1. confidentiality: Eve should not be able to get the plaintext
2. data integrity: Bob should be sure that Alice's message has not been altered.
3. authentication: Bob should be sure that the message comes from Alice.
4. non-repudiation: Alice cannot claim that she did not send the message.

some crypto protocols

1. digital signatures - how to sign electronically
2. identification - with password but also zero-knowledge
3. key exchange - how Alice and Bob can agree on a key
4. secret sharing - n parties, each one has a share of a secret; they can reconstruct the secret only if all of them put their shares together.
5. electronic cash - paying anonymously, unlike the use of credit cards
6. games - how to flip a coin by e-mail, ...