Cryptography & Digital Signatures

CS 594 Special Topics/Kent Law School:

Computer and Network Privacy and Security: Ethical, Legal, and Technical Consideration

Cryptography

- The science of sending secret messages
- Ancient history; people always interested in it
  - Mentioned in Herodotus; the Hebrew Bible

Overview

- Uses: Secrecy, nonrepudiation, authentication
- Implementations:
  - Rotor machines (Haeglin, Enigma)
  - Computers, special-purpose chips, etc.

Most basic scenario

- Alice uses ***encryption*** algorithm E to transform her message m, the ***plaintext*** (or ***cleartext***), and a key k into ***ciphertext*** E(m,k).
- Intended recipient has key that allows him to ***decrypt*** the ciphertext E(m,k) and get back m.

Crypto 1800–1975

- In past century or two, secrecy rests upon secret key. I.e., ciphertext can be decrypted by anybody possessing (or guessing) the secret key.
- Before modern era (c. 1976–), security rests on some sort of mixing and can be broken with enough samples by statistical techniques (with exception of one-time pad)

Aside: Breaking Enigma

- Family of German rotor machines.
- Commercial originally; military Wehrmacht version is the famous one.

Breaking The Unbreakable

- 1932: Trio of Polish mathematicians led by Marian Rejewskibroke 3-rotor plus plugboard machine

Enigma continued

- 1939: Germans went up to 5 rotors; more than Polish system could handle
- July 1939 Polish mathematicians gave their techniques to French & British
- September 1939, Turing at Blechley Park begins effort to build Bombe to decrypt this Enigma—and succeeds!

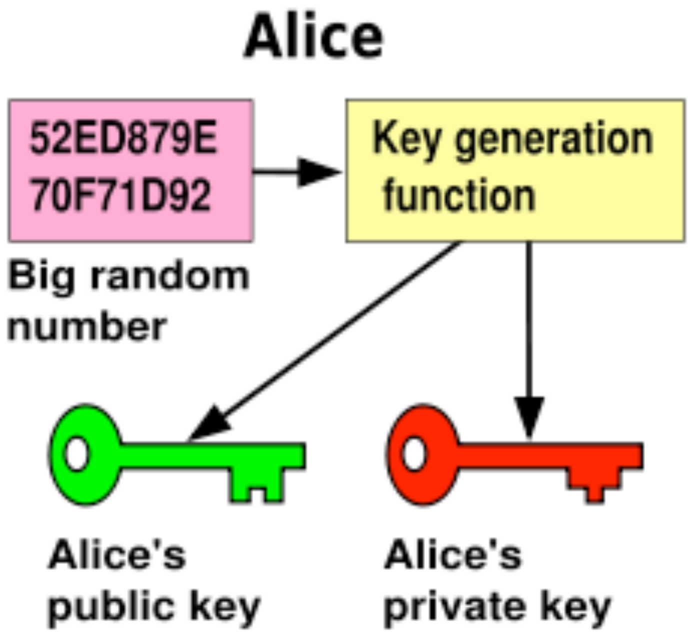
Modern era: 1976–

- Cryptography based on (computational) complexity theory—theory of what can be computed quickly versus what can be computed slowly
- Goal: encryption, and decryption *with possession of proper key* can be computed very fast; decryption without key is *very* slow (e.g., 1 million years on best supercomputer).
- Currently: True if make unproven assumptions overwhelmingly believed by researchers in complexity theory. (E.g., “Factoring not in P”.)

Two kinds of crypto

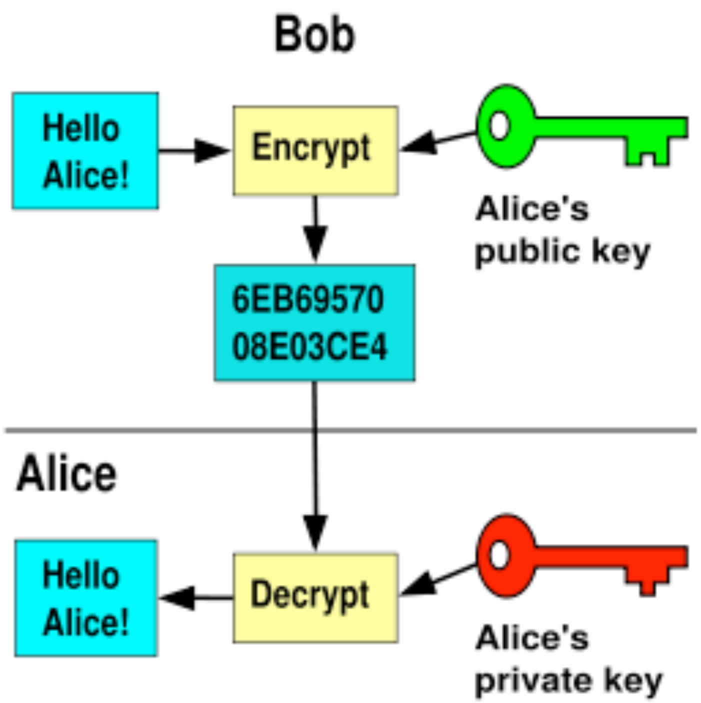
- **Symmetric** or **secret key:** there is a unique key, and Alice and Bob must somehow arrange to share it so they but only they know it.
  - In practice, very fast encrypt & decrypt
  - Only kind of crypto prior to 1976.
- **Asymmetric** or **public key:** Each user has 2 keys: secret one to decrypt; *public key* that anybody can use to send her messages. Medium speed in practice.

Public-key cryptography: 1



- A big random (i.e., pesudorandom) number is used to make a *key pair*.

Public-key cryptography: 2



- Anyone can encrypt using public key; can decrypt only with private key
- Often encrypt & decrypt same function; only keys different

Some well-regarded public-key methods

- RSA
- Digital Signature Standard/Algorithm (DSS)/DSA (NIST standard; signatures only)
- El Gamal
- Various elliptic curve methods

Public-key in practice

- Because somewhat slower, used mostly for only two things:
  1. Digital signatures
  2. Various techniques having to do with key management & distribution (more soon)

Well-known private-key crypto

- Data Encryption System (DES): Standard from federal government in 1977. Fatal flaw for 21st century: 56-bit key length subject to brute-force attacks.
  - Still in use though!
- Triple DES; longer key length; popular stopgap as DES became scary.
- Advanced Encryption Standard (AES): newish standard from federal government; very fast, very secure; seeing slow conversion from legacy systems.

One-time pad

- Special case of secret key. Key length must be equal to message length (huge drawback).
- Provides perfect secrecy, with no assumptions whatsoever.
- Believed to be used for high-level diplomatic and intelligence work; may become more prevalent

Crypto Goals: confidentiality

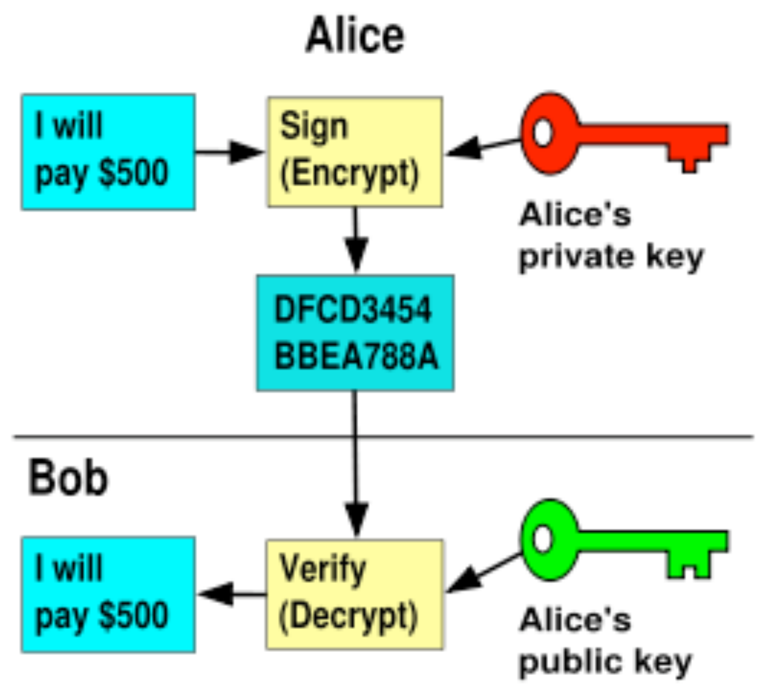
- The obvious one. Use the cryptosystem.
- Advantage of public-key cryptography is that it allows for secrecy between two parties who have not arranged in advance to have a shared key (or trusted some third party to give it to them).
- Disadvantage is speed. Therefore, in practice, hybrid systems–use public-key to establish session key for private key.

Goals: integrity

- Simplest form is an elaboration of the checksum: **(cryptographic) hash function** or **message digest.**
  - Short “signature” of the message, 128–512 bits that depend on entire message; *extremely* improbable that unequal messages have same hash.
  - Popular functions: SHA-1 (weak?), SHA-256, SHA-384, SHA-512 (NIST); MD5 (Rivest)
  - By itself, shows only no accidental corruption

Digital Signatures

- Use public-key cryptography “backwards” to sign messages.
- I.e., to sign m, Alice encrypts m with her *private key*; anybody can verify by using Alice’s public key.



Non-repudiation + integrity

- A’s digital signature of a cryptographic hash of message m guarantees that m was signed by A and that m was not altered.
- Anybody can computer the hash of m; anybody can verify A’s signature.
- Or conceptually more complex Message Authentication Code (MAC)

How do I get Alice's public key?

- Even with public-key crypto and digital signatures, still have the problem of authentication: binding users to keys.
- Early days articles envisioned phonebook-like database with Name, Public Key entries
- Problem: How secure is that database?!
  - Attacker can put in his own key for me, and start signing contracts (and checks!) in my name.
  - Maybe we can secure the phonebook, but then that kills the idea of of keys widely, easily *publicly* available.

Certification

- Common solution today is for trusted 3rd party—**certification authority (CA)** to sign the user's public encryption key.
- Resulting **certificate** will contain, e.g., user's name/ID, user's public key; CA's name; certificate's start date, and length of time it is valid.
- User publishes certificate
- Standard X.509 for format of certificate

Web of trust

- Alternative to CA is web-of-trust model of PGP, GnuPG, and OpenPGP.
- Key-signing party!

Encryption of Stored Data

- Routine to use encryption for transmission over network today; but in practice need more encryption of "*data at rest*"—data stored locally.
  - December 2004 Bank of America backup tapes with cleartext 1.2M government employees name, SSN, bank account #
  - April 2005, San Jose Medical group had computer physically stolen: 200K patient

Encrypting data at rest

- Simple solution is to encrypt data at rest
- For laptops can use commercially available encryption package such as PGP
- Also now back-end appliances between data & backup device, various other products.