Lecture 19
Cryptography

Bundesarchiv, Bild 101I-769-0229-11A
Foto: Borchert, Erich (Eric) | 1940 Mai - Juni

Cryptography

- **some history**
  - Caesar cipher, rot13
  - substitution ciphers, etc.
  - Enigma (Turing)
- **modern secret key cryptography**
  - DES, AES
- **public key cryptography**
  - RSA, digital signatures, cryptographic hashing
- **cryptography in practice**
  - e-commerce
  - Tor browser
  - Bitcoin
  - politics

Cryptography basics

- **Alice & Bob want to exchange messages**
  - keeping the content secret
  - though not the fact that they are communicating
- **they need some kind of secret that scrambles messages**
  - makes them unintelligible to bad guys but intelligible to good guys
- **the secret is a "key" (like a password)**
  - known only to the communicating parties
  - that is used to do the scrambling and unscrambling

  - for Caesar cipher, the "key" is the amount of the shift (A => D, etc.)
  - for substitution ciphers, the key is the permutation of the alphabet
  - for Enigma, key is wiring and position of wheels plus settings of patches

  - for modern ciphers, the key is a large integer used as part of an intricate algorithmic operation on the bits of the message

Modern secret key cryptography

- **messages encrypted and decrypted with a shared secret key**
  - usually the same key for both operations ("symmetric")
- **encryption/decryption algorithm is known to adversaries**
  - "security by obscurity" *does not work*
- **attacks**
  - decrypt specific message(s) by analysis
    - various combinations of known or chosen plaintext and ciphertext
  - determine key by "brute force" (try all possible keys)
- **if key is compromised, all past and future messages are compromised**
- **big problem: key distribution**
  - need a secure way to get the key to both/all parties
    - diplomatic pouches, secret agents, …
  - doesn't work when the parties don't know each other
  - or have no possible channel for exchanging a secret key
  - or when want to exchange secret messages with many different parties
    - e.g., credit card numbers on Internet

The secrecy is in the key

"Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi."

(The system must not require secrecy, and it does not matter if it falls into the hands of the enemy.)

Auguste Kerckhoffs, "La cryptographie militaire",
Journal des sciences militaires, vol IX, pp 5-38, Janvier 1883.

- we have to assume that the bad guys know how the encryption and decryption operate
- “security by obscurity” does not work
- Claude Shannon: "The enemy knows the system."

DES and AES

- **Data Encryption Standard (DES)**
  - developed ~1977 by IBM, with NSA involvement
  - widely used, though lingering concerns about trap doors
  - by 1995, 56-bit key was much too short:

    could exhaustively test all keys in a few hours with special hardware
  - "triple DES" used 3 DES encryptions to increase effective key length

    but not enough to prevent brute-force attacks

- **Advanced Encryption Standard (AES)**
  - result of an international competition run by NIST (www.nist.gov/aes)
  - completely open: algorithms and analyses in public domain
  - Rijndael: winning algorithm selected October 2000

    approved as official US government standard
  - 128, 192, 256-bit keys
  - fast in both hardware and software implementations

Public key cryptography

- a fundamentally new idea
  - Diffie & Hellman (USA, 1976); invented earlier in England but kept secret
- each person has a (public key, private key) pair
  - the public and private keys are mathematically related
  - a message encrypted with one key can only be decrypted with the other key
- public key is published, visible to everyone
- private key is secret, known only to owner

- Alice sends a secret message to Bob by
  - encrypting it with *Bob's* public key
  - only Bob can decrypt it, using his private key
- Bob sends a secret reply to Alice by
  - encrypting it with *Alice's* public key
  - only Alice can decrypt it, using her private key
- Eve knows Alice and Bob are talking
  - but can't decrypt what they are saying

RSA public key cryptographic algorithm

- **most widely used public key system**
- **invented by Ron Rivest, Adi Shamir, Len Adleman, 1977**
  - patent expired Sept 2000, now in public domain

- **based on (apparent) difficulty of factoring very large integers**
  - "large" >= 1024 bits ~ 300 digits
  - public key based on product of two large (secret) primes
  - encrypting and decrypting require knowledge of the factors
- **slow, so usually use RSA to exchange a secret "session key"**
  - session key used for secret key encryption with AES
  - used by SSH for secure login
  - used by browsers for secure exchange of credit card numbers

    https: http with encryption
  - SSL (Secure Sockets Layer) or TLS (Transport Layer Security)

    used to encrypt TCP/IP

Intuition for public-key crypto algorithms

- one-way function: easy to compute, hard to invert
  - e.g, multiplication vs factoring
- need a trap door in one-way function that makes inverse easy
- if you know something extra
  - e.g., one of the factors
  - then you can decrypt easily
- if you don't know the something extra
  - you can't decrypt easily

Digital signatures

- **can use public key cryptography for digital signatures**
  - if Alice encrypts a message with her <u>private</u> key
  - and it decodes properly with her public key
  - it had to be Alice who encoded it
- **signature can be attached to a message**
  - Alice encrypts a message with her private key
  - Alice encrypts the result with Bob's public key
  - only Bob can decrypt this (with his private key)

    but it won't make any sense yet
  - Bob then decrypts it with Alice's public key
  - if it decodes properly, it had to be Alice who encrypted it originally
- **necessary properties of digital signatures**
  - can only be done by the right person: can't be forged
  - can't re-use a signature to sign something else
  - signature attached to a document: signs specific contents
  - signature can't be repudiated

Cryptographic / secure hashing

- **digital signature usually done by signing a ”secure hash" or “message digest” of a document, not the document itself**
- **secure hash algorithm reduces input data to a comparatively short number such that**
  - any change to the original document produces a completely different hash
  - can't deduce the original document from the number
  - can't find another document that has the same hash
- **current secure hash algorithms**
  - MD5 (Rivest, MIT): 128 bits
  - SHA-1 (US government standard): 160 bits
  - SHA-2 (also standard): family of 224, 256, 384, or 512 bits
- **international competition to create a new secure hash, SHA-3**
  - analogous to AES competition (also run by NIST)
  - first round submissions 10/08, final round 12/10,
  - winner announced in Oct 2012; official 2015

Properties of public/private keys

- **can't deduce the public key from the private, or vice versa**
- **can't find another encryption key that works with the decryption key**
- **keys are long enough that brute force search is infeasible**

- **nasty problems:**
  - if a key is lost, all messages and signatures are lost
  - if a key is compromised, all messages and signatures are compromised
  - it's hard to revoke a key
  - it's hard to repudiate a key (and hard to distinguish that from revoking)

- **authentication**
  - how do you know who you are talking to? is that really Alice's public key?
  - public key infrastructure, web of trust, digital certificates