Secure Communications

Fred Long

"Applied Cryptography" by Bruce Schneier,
John Wiley & Sons Inc., 2e, 1996.
ISBN 0-471-12845-7

"The Code Book" by Simon Singh,
Fourth Estate Limited, 1999.
ISBN 1-85702-879-1

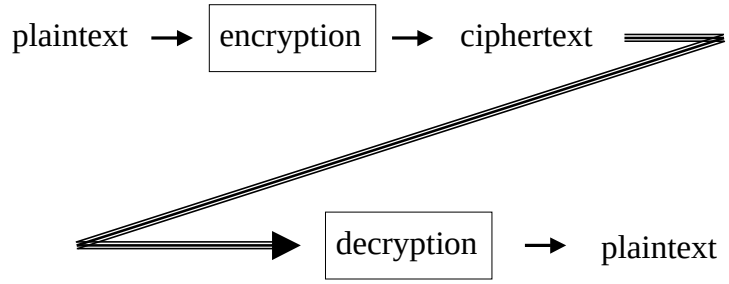
"Web Security & Commerce" by Simson Garfinkel with Gene Spafford,
O'Reilly & Associates, Inc., 1997.
ISBN 1-56592-269-7

Problems:

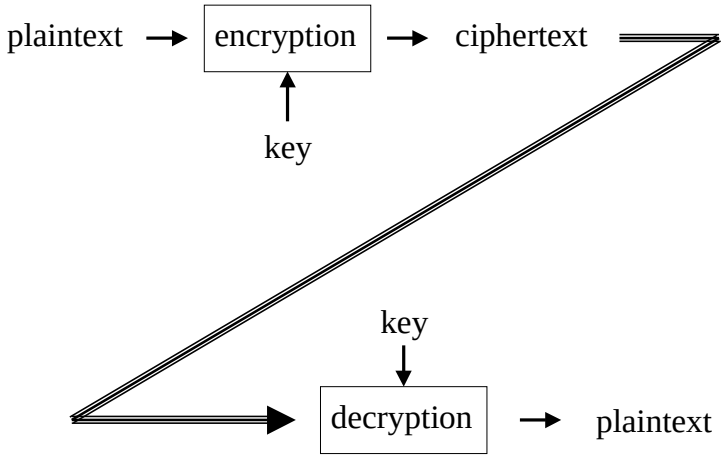
- Confidentiality
- Identification and Authentication
- Data Integrity
- Non-repudiation
- Authorisation

Encryption

- Encryption involves scrambling a message



- There are just a few encryption/decryption algorithms
- Therefore, the security of the encryption must depend on a *parameter* of the encryption, not just on the algorithm
- This parameter is called the *key*



For example, a Caesar Cipher:

`abcdefghi...rstuvwxyz`

`defghijkl...uvwxyzabc`

So, **`fred -> iuhg`**

In this case, the key is 3

- If the encryption and decryption keys are the same, we talk of *symmetric key cryptography*
- Otherwise, we talk of *asymmetric key cryptography*, or *public key cryptography*
- Practical asymmetric key cryptographic methods were first discovered in the UK in the early 1970s but kept secret; the first published methods appeared in the US in the late 1970s

- The work in the UK was carried out at the Government Communications Headquarters (GCHQ) by:
  - James Ellis
  - Malcolm Williamson
  - Clifford Cocks
- The papers are now available on the web at: http://www.cesg.gov.uk/about/nsecret.htm

Symmetric Key Cryptography

- Data Encryption Standard (DES); 1977, National Bureau of Standards (US); 56-bit fixed length key; very efficient, particularly if special hardware is used
- Triple-DES (or DES-EDE) uses three applications of DES with two, independent DES keys; an effective key length of 112-bits

- DES is a *block cipher*; it encrypts data in 64-bit blocks
- The 64-bit block is split into left and right halves, then there are 16 rounds of identical operations:

  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

  where K_i is derived from the initial key by shifting the bits
- The function *f* involves shifting, substituting and XORing bits

- International Data Encryption Algorithm (IDEA); James Massey and Xuejia Lai, ETH Zurich, 1991; patented by the Swiss Ascom Tech AG; block cipher; fixed length 128-bit key; faster than triple-DES
- RC2 and RC4; Ron Rivest, RSA Data Security Inc., early 1990s; variable length keys; claimed to be faster than IDEA
- Blowfish; Bruce Schneier, 1994; public domain algorithm; variable length key; compact, simple and fast

Key size does matter:

Size	No. keys	Time to crack	When
40-bits	$\approx 1.1\times10^{12}$	3.5 hrs	Jan. '97
48-bits	$\approx 2.8\times10^{14}$	313 hrs	Feb. '97
56-bits	$\approx 7.2\times10^{16}$	22 hrs	Jan. '99

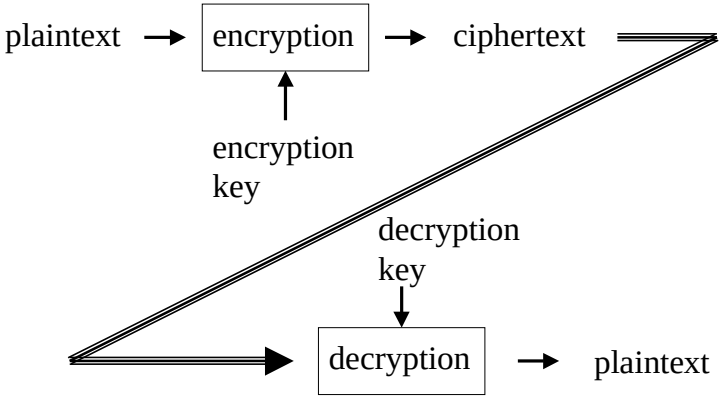
- The big difficulty with symmetric key cryptography is the *key management problem*:
  - Since the encryption and decryption keys are the same, they must be communicated securely
  - Key changes must be synchronised between sender and recipient
- So, symmetric key cryptography amounts to securely communicating a small piece of data (the key) in order to be able to then securely communicate large pieces of data (the messages)

Diffie-Hellman Key Exchange

- In 1976, Whitfield Diffie and Martin Hellman published a key exchange protocol that overcame the key management problem
- Suppose Alice and Bob want to set up a secure key
- They find one-way functions *A*, *B* such that $A(B(y))=B(A(y))\ \forall y$; then they agree on *x*; Alice sends Bob *A(x)*; Bob sends Alice *B(x)* and they use the key $k=A(B(x))=B(A(x))$

- In Diffie-Hellman, $A(x) = x^a \bmod p$, $B(x) = x^b \bmod p$, where *p* is a large prime number
- So, $k = A(B(x)) = (x^b)^a \bmod p = x^{ba} \bmod p = x^{ab} \bmod p = B(A(x))$
- The number *a* is known only to Alice, the number *b* is known only to Bob, the numbers *p*, *x*, *A(x)*, *B(x)* can be made public
- Calculating *k* given *x* involves finding the values *a* and *b* which means solving the discrete logarithm problem

Public Key Cryptography



- The encryption and decryption keys are, clearly, related
- However, given the encryption key, it is difficult to calculate the decryption key
- For example, the encryption key can involve knowledge of the product of two very large prime numbers while the decryption key requires knowledge of the two separate prime numbers (factorising large numbers is notoriously difficult)
- So, the encryption key can be made public while the decryption key remains private

- In RSA, we find two very large primes p, q and a positive integer e invertible modulo $(p-1)(q-1)$ so, we can find d such that
  $de \equiv 1 \pmod{(p-1)(q-1)}$
- d can be found easily if we know p and q but not if we know only $n = pq$
- The encryption key is the pair (n, e):
  $E(b)$ = residue of b^e modulo n
- The decryption key is the pair (n, d):
  $D(a)$ = residue of a^d module n
- $D(E(b)) = b$ since $b^{(p-1)(q-1)} \equiv 1 \pmod{n}$

- Note that public key encryption methods can work the other way round, i.e., a message encrypted with the private key can be decrypted with the public key
- However, a message encrypted with the private key is not confidential, because anyone can access the public key
- Nevertheless, we will see applications of this back-to-front use of public key encryption later

- RSA, named after its inventors: Ronald Rivest, Adi Shamir, and Leonard Adleman, 1978; much more computationally intensive than symmetric key algorithms (e.g., RSA is 100 times slower than DES in software, up to 1000 times slower in hardware)
- ElGamal; El Gamal, 1985; depends on an arithmetic algorithm different from that of RSA (the so-called discrete logarithm problem)
- Elliptic curve cryptosystems; proposed by Koblitz and Miller (independently) in 1985; more efficient than RSA

One-Way Hashing

- A *hash function*, *H(M)*, takes an arbitrary length message *M* and produces a fixed length hash value *h = H(M)*.
- A one-way (or secure) hash function has the properties that:
  - given *h*, it is difficult to calculate *M*
  - given *M*, it is difficult to find another *M'* such that *H(M) = H(M')*

- One-way hash functions are used to secure passwords; the hash of the password is stored, not the password itself
- They also play a role in secure communications
- MD2, MD4, MD5; these are *message digest* methods devised by Ron Rivest; they produce 128-bit hashes
- Secure Hash Algorithm (SHA); developed by the National Institute of Standards and Technology and the National Security Agency in the US; produces a 160-bit hash

Confidentiality

- Clearly, any encryption method can guarantee confidentiality
- The key management problem can be overcome by using either Diffie-Hellman key exchange or a public key encryption method

Identification and Authentication

- If someone encrypts a message (almost any message) with their private key then anyone else can be sure of the originator of the message, because only they have access to their private key
- Hence, this back-to-front use of public key encryption can be used to establish identification

- A service wishing to be certain that it is communicating with who it thinks it is can send the communicating party a random message and ask the party to return it encrypted with the party's private key
- If the returned message successfully decrypts with the communicating party's public key then the service can be sure that the party is genuine
- Hence, back-to-front public key encryption also leads to authentication

Data Integrity

- It is often necessary to ensure that data has not been corrupted or modified during transmission over a network
- This can be done by generating a hash of the original data and comparing a hash of the received data with the original hash
- For security, a secure hash can be used and the original hash sent encrypted by the sender's private key

Non-repudiation

- If a message is "signed" with a secure hash encrypted using the sender's private key then the sender cannot later claim that they did not send that message
- Similarly, if the receiver returns a secure hash encrypted with their private key then they cannot later deny receiving the message

Authorisation

- Authorisation is concerned with granting privileges depending on the identity of users and agents
- So, this is clearly related to identification and authentication
- But, if we receive a public key, how can we be sure it is from who it claims to be from?

Digital Certificates

- Possession of a public key ensures only that the holder of the corresponding private key can decrypt the message
- It does not guarantee the identity of the private key holder
- Trust is achieved only after the public key is bound to a known identity
- A digital certificate binds the identity of an entity (person or system) to their public key

- A digital certificate is, essentially, the entity's public key (plus, possibly, other information) signed with the private key of the known authority (called the *certificate authority* or CA)
- Common types of certificate:
  - personal certificates
    - identifies an individual for authorisation, signing, ...
  - site certificate
    - identifies a system for, e.g., secure communication
  - code-signing certificate
    - identifies the developer to guarantee authenticity, ...
  - authority certificate
    - identifies the CA, used to sign other certificates

Certificate Chains

- Essentially, a certificate contains the subject's and the issuer's data — the subject is chained to the issuer
- Why should one trust the CA who signs a certificate?
- If necessary, the subject–issuer chain can be continued

subject: Fred Long
issuer: Dept. Comp. Sci.
subject: Dept. Comp. Sci.
issuer: UWA
subject: UWA
issuer: HEFCW
subject: HEFCW
issuer: Welsh Assembly
subject: Welsh Assembly
issuer: UK Government
subject: UK Government
issuer: UK Government

- There are international standards for digital certificates, e.g., X.509
- Example CAs:
  - VeriSign: http://www.verisign.com/
  - Thawte: http://www.thawte.com/ (although Thawte has been taken over by VeriSign)
- Certificates from VeriSign and Thawte are recognised by the main browsers
- One can generate one's own certificates using (e.g., Netscape's signtool, MS's certificate server, Java's keytool) but these will not be recognised by the browsers

Secure Socket Layer (SSL)

- SSL was developed by Netscape in 1994. The latest version is version 3.0.
- SSL is a *protocol* that enables the client and server to agree on the cryptographic techniques they will use, authenticate each other, and then exchange a master key for use in subsequent communications

Transport Layer Security (TLS)

- The Internet Engineering Task Force (IETF) is attempting to introduce an international standard, based on SSL 3.0, called Transport Layer Security (TLS)
- TLS is similar to SSL 3.0, but has a slightly different cipher suite and uses the HMAC secure message digest instead of MD5
- A Request for Comments on the TLS Protocol v 1.0 was issued in January 1999

Pretty Good Privacy (PGP)

- PGP was developed by Philip Zimmermann in 1991 as an electronic-mail security program
- PGP uses RSA for key exchange and digital signatures; MD5 for integrity checking; ZIP for compression; and IDEA for data encryption
- PGP uses “rings of trust” instead of chains

Privacy Enhanced Mail (PEM)

- PEM is another IETF standard
- Again, it is really a protocol
- PEM allows for
  - Data encryption: DES
  - Key management: DES or RSA
  - Integrity checking: RSA/MD2 or RSA/MD5
  - Digital signing: RSA/MD2 or RSA/MD5