BSc in Computer Engineering
CMP4103
Computer Systems and Network Security

Lecture 3

Cryptography

Eng Diarmuid O'Briain, CEng, CISSP

Department of Electrical and Computer Engineering,
College of Engineering, Design, Art and Technology,
Makerere University

Copyright © 2017 Diarmuid Ó Briain

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Table of Contents

1. History of Cryptography

1.1 Classic Cryptography

The earliest known use of cryptography is found in non-standard hieroglyphs carved into monuments from Egypt. Some clay tablets from Mesopotamia, somewhat later are clearly meant to protect and even later on the timeline Hebrew scholars made use of simple monoalphabetic substitution ciphers beginning perhaps around 500 to 600 BC.

1.1.1 Atbash cipher

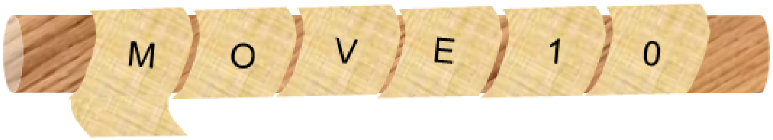
This cipher mentioned above is the Atbash cipher, it is a simple ***substitution cipher*** for the Hebrew alphabet. It consists in substituting the first letter for the last, the second for the one before last etc, reversing the alphabet. Here is an example using the standard English alphabet.

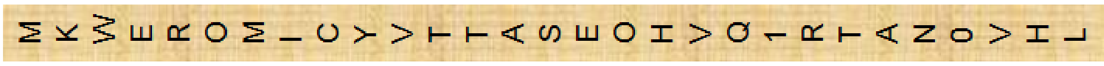
A | B | C | D | E | F | G | H | I | J | K | L | M

Z | Y | X | W | V | U | T | S | R | Q | P | O | N

1.1.2 The Scytale

The Greeks of Classical times are said to have known of ciphers like the scytale ***transposition cipher*** claimed to have been used by the Spartan military. The system works by the use of a cylinder with a strip of leather or paper wound around it on which a message is written.





The recipient uses a rod of the same diameter on which he wraps the leather or paper to read the message. It has the advantage of being fast and not prone to mistakes. It can, however, be easily broken. Figure out what the message reads.

1.1.3 Polybius Square

Another Greek method was developed by Polybius and is known as the Polybius Square or Checkerboard. Each letter is then represented by its coordinates in the grid. For example, "CISSP" becomes "13 23 41 41 34".

Encrypt the line “Polybius was a greek crypto master”.

	1	**2**	**3**	**4**	**5**	**6**	
1	A	B	C	D	E	F	**1**
2	G	H	I	J	K	L	**2**
3	M	N	O	P	Q	R	**3**
4	S	T	U	V	W	X	**4**
5	Y	Z	0	1	2	3	**5**
6	4	5	6	7	8	9	**6**
	1	**2**	**3**	**4**	**5**	**6**	

1.1.4 Caesar cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

The Romans certainly did know something of cryptography. Here we see the Caesar’s cipher this is a type of ***substitution cipher*** in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. This is also called the ROT3 (ROTate 3) as it rotates by 3 places.

Encrypt the line “*Caesar was a Roman Emperor*”.

1.2 Modern age cryptography

1.2.1 Enigma machine

The first Enigma was invented by German engineer Arthur Scherbius at the end of World War I. This model and its variants were used commercially from the early 1920s, and adopted by military and government services of several countries most notably by Nazi Germany before and during World War II.

The Enigma machine is a combination of mechanical and electrical subsystems. The mechanical subsystem consists of a keyboard; a set of rotating disks called rotors arranged adjacently along a spindle; and one of various stepping components to turn one or more of the rotors with each key press. The stepping component varies slightly from model to model. Most often the right-hand rotor steps once with each key stroke, and other rotors step occasionally. The continual movement of the rotors results in a different cryptographic substitution after each key press.

Polish Military Intelligence reconstructed the German Enigma machine and techniques for decrypting ciphers produced on it and presented it to their French and British allies in Warsaw on July 26, 1939, just five weeks before the start of WWII. The British in Bletchley Park created a team of code breakers under a project called ULTRA were able to decrypt a vast number of messages which had been enciphered using the Enigma.

1.2.2 Red and Purple machines

“System 91 Printing Machine” or “Type A Cipher Machine” was codenamed RED by the US, was a diplomatic cryptographic machine used by the Japanese Foreign Office before and during World War II. A relatively simple device where encryption was provided through a single half-rotor, it was quickly broken by western cryptographers.

The RED cipher was succeeded by the “System 97 Printing Machine for European Characters” or “Type B Cipher Machine” and was codenamed PURPLE by the US, it was a diplomatic cryptographic machine used by the Japanese Foreign Office just before and during World War II. The machine was an electromechanical stepping-switch device.

The PURPLE machine was more secure than Red, but the Imperial Japanese Navy did not recognise that RED had already been broken. The PURPLE machine inherited a weak point from the RED machine, namely vowel-consonant separate encryption, which was called "sixes-twenties" by the US Army Signals Intelligence Service (SIS).

2. Cryptographic Basics

Cryptography or cryptology is the practice and study of hiding information. Modern cryptography intersects the disciplines of mathematics, computer science, and engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

2.1 Goals of Cryptography

The goals of cryptography are data privacy or confidentiality, data authenticity to prove it came from where it claims to come from and data integrity to demonstrate that it has not been modified on the way during transmission. Non-repudiation is a goal that is achieved by a combination of integrity and authentication. It is an assurance that the message originated from the sender and not someone masquerading as the sender.

- Confidentiality
- Integrity
- Authentication
- Non-repudiation

2.2 Kerckhoffs' principle

Kerckhoffs' principle or Kerckhoffs' law from Auguste Kerckhoffs a Paris based Dutch linguist and cryptographer in the 19th century stated that:

"*a cryptosystem should be secure even if everything about the system, except the key, is public knowledge*"

Kerckhoffs' principle was reformulated by Claude Shannon an American electronic engineer and mathematician as Shannon's' maxim:

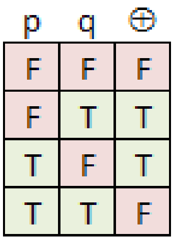
"*The enemy knows the system*"

The law was one of six design principles laid down by Kerckhoffs for military ciphers. Translated from French, they are:

1. The system must be practically, if not mathematically, indecipherable.
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents.
4. It must be applicable to telegraphic correspondence.
5. It must be portable, and its usage and function must not require the concourse of several people.
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

3. Some Mathematics

3.1 eXclusive OR



p	q	⊕
F	F	F
F	T	T
T	F	T
T	T	F

The logical operation exclusive OR (XOR) is a type of logical disjunction on two operands that results in a value of true if exactly one of the operands has a value of true. A simple way to state this is "one or the other but not both."

The truth table on the left shows the outcome for p XOR q which is usually written as p ⊕ q for each value of p and q.

Exclusive-or is sometimes used as a simple mixing function in cryptography, for example, with one-time pad or block ciphers as we shall see later in this lecture.

3.2 Modulo

The modulo operation finds the remainder of division of one number by another.

Given two numbers, *a* (the dividend) and *n* (the divisor), a modulo *n* is the remainder, on division of *a* by *n*. For example, the expression "*7 mod 3*" would evaluate to *1*, while "*9 mod 3*" would evaluate to *0*.

4. Other Cryptographic terms

4.1 One way function

A one-way function is a mathematical function that is easy to compute an output for every input but practically impossible to determine the input given the output and the function.

A trapdoor one-way function or trapdoor permutation is a special kind of one-way function. Such a function is hard to invert unless some secret information, called the trapdoor, is known. RSA is a well known example of a function believed to belong to this class.

4.2 Confusion and Diffusion

These are two properties of the operation of a secure cipher which were identified by Claude Shannon in his paper Communication Theory of Secrecy Systems, published in 1949.

Confusion refers to making the relationship between the key and the ciphertext as complex and involved as possible. This prevents the attacker altering plaintext and analysing the result to determine the key.

Diffusion means that the output bits should depend on the input bits in a very complex way. In a cipher with good diffusion, if one bit of the plaintext is changed, then the resultant ciphertext should have multiple changes spread right across it, i.e. a complete change.

4.3 nonce

A nonce is an abbreviation of number used once. It is typically a random or pseudo-random number issued in a security protocol to ensure that past communications cannot be reused in replay attacks.

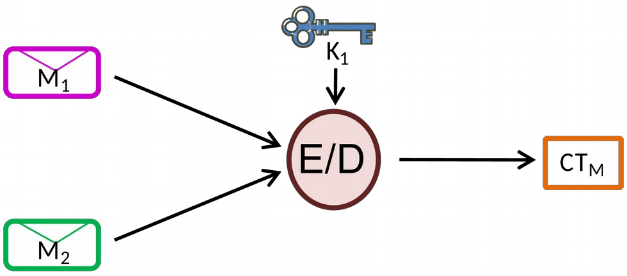
4.4 Initialisation Vector (IV)

An initialisation vector (IV) is a block of bits that is required to allow a stream cipher or a block cipher to be executed in any of several streaming modes of operation to produce a unique stream independent from other streams produced by the same encryption key, without having to go through a (usually lengthy) re-keying process.

4.5 Work Factor

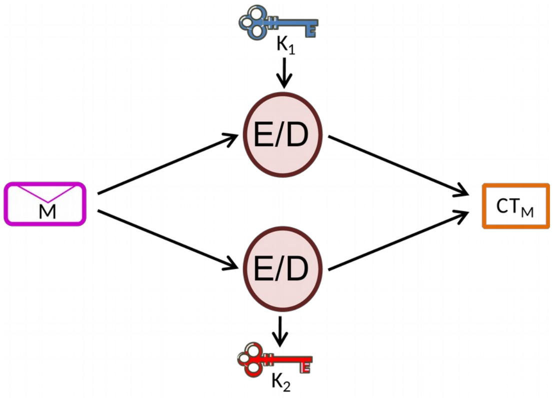
Compare the cost of circumventing the mechanism with the resources of a potential attacker. The cost of circumventing, commonly known as the "work factor", in some cases can be easily calculated. It is often considered as the time or effort required to perform a brute force attack against a cryptosystem.

4.6 Collision



A collision is said to have occurred when two different messages through the same cryptosystem with the same key gives the same ciphertext output.

4.7 Key Clustering



Key clustering occurs when two different keys generate the same ciphertext from the same plaintext, using the same cipher algorithm. A good cipher algorithm, using different keys on the same plaintext, should generate a different ciphertext, irrespective of the key length.

4.8 Codes and Ciphers

A code replaces words, phrases, or sentences with groups of letters or numbers. This need not always be for authentication for example in radio voice procedure the proword “*over*” means “*I have completed my transmission and expect a reply from you*”, in this case “*over*” is a code.

A cipher rearranges letters or uses substitutes to disguise the message.

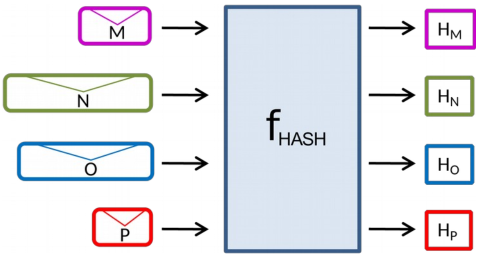
4.9 One-time pad

The one-time pad (OTP) is the only type of encryption that has been proven to be absolutely impossible to crack if used correctly. The plaintext is encrypted with a substitution cipher using a secret random key (or pad) as long as the plaintext, resulting in a ciphertext of random data. If the key is truly random, as large as the plaintext, never reused in whole or part, and kept secret, the ciphertext will be impossible to decrypt or break without knowing the key.

The "pad" part of the name comes from early implementations where the key material was distributed as a pad of paper, so the top sheet could be easily torn off and destroyed after use.

The OTP is also known as the Vernam Cipher after inventor Gilbert Stanford Vernam.

4.10 Hash Function



A hash function is a mathematical function that converts a large, possibly variable-sized amount of data into a small fixed size message digest (hash) that can.

The hash function has four properties:

1. Easy to compute the hash value for any message.
2. Infeasible to find a message that has already a given hash.
3. Infeasible to modify a message without a change to its hash.
4. Infeasible to find two different messages with the same hash.

4.10.1 Message-Digest algorithm 5 (MD5)

MD5 is a cryptographic hash function designed by Professor Ronald Rivest in 1991. It is published as RFC 1321. It has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. MD5 uses a 128-bit hash value and the hash is typically expressed as a 32-digit hexadecimal number. A number of flaws have been found from as far back as 1996 however in 2007 it was described how to create a pair of files that share the same MD5 checksum which would allow the generation of fake valid SSL certificates. The U. S. Department of Homeland Security said MD5 "*should be considered cryptographically broken and unsuitable for further use*".

4.10.2 Secure Hash Algorithm (SHA)

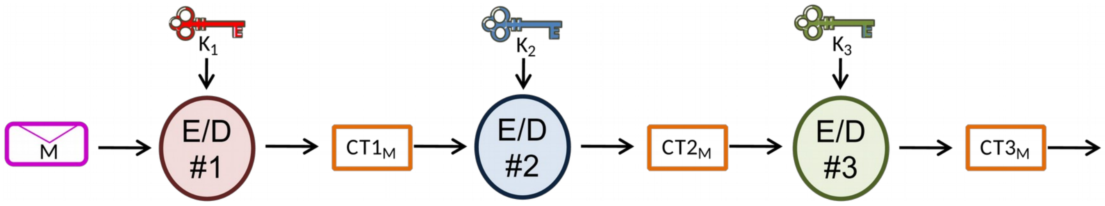
A popular alternative to MD5 has been SHA-1 (RFC 3174). The SHA-1 algorithm is based on principles of MD4 (an earlier Ronald Rivest hash algorithm that MD5 is also based on). It was published by National Institute of Standards and Technology (NIST) as FIPS PUB 180-1 in 1995. SHA-1 produces a 160-bit digest from a message with a maximum length of ($2^{64} - 1$) bits.

In 2001 NIST published FIPS PUB 180-2. This extended SHA-1 with further versions to be collectively called SHA2. These were SHA-224, SHA-256, SHA-384, and SHA-512 named after their digest lengths. The US agencies were required to stop most use of SHA-1 after 2010. In October 2012, Keccak was selected as the winner of the NIST hash function competition, SHA-3. As SHA-2 had not been broken it is not a replacement but NIST perceived the need for an alternative, dissimilar cryptographic hash and in August 2015 was published as FIPS 202 "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions". This includes 4 hash and 3 eXtendible Output Functions (XOF). XOF functions are different from hash functions, but it is possible to use them in similar ways, with the flexibility to be adapted directly to the requirements of individual applications.

4.10.3 Hash Function summary

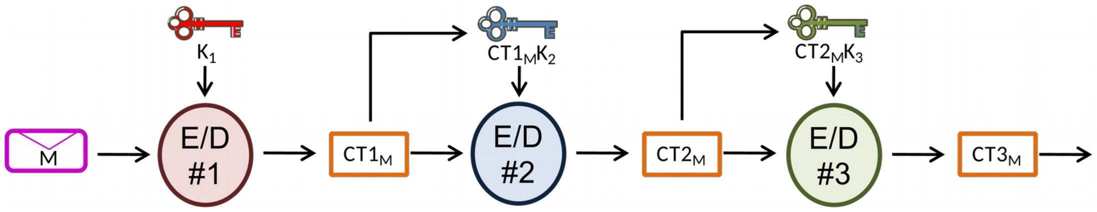
Algorithm	Name	Type	Block Size	Keys	Rounds	Other
SHA-1	Secure Hash Algorithm	Hash	512	160	80	160 bits output - 80 bits protection against collision
SHA-256	Secure Hash Algorithm	Hash	512	256	64	256 bits output - 128 bits protection against collision
SHA-512	Secure Hash Algorithm	Hash	Variable	512	80	512 bits output - 256 bits protection against collision
SHA3-256	Secure Hash Algorithm	Hash	576	1600	24	
SHAKE-128	Extendable Output function	XOF	1344	1600	24	
SHAKE-256	Extendable Output function	XOF	1088	1600	24	
MD4	Message-Digest algorithm 4	Hash	Variable	128	3	128 bits output
MD5	Message-Digest algorithm 5	Hash	Variable	128	4 (each round is composed of 16 similar operations)	128 bits output

4.11 Synchronous Cryptosystem



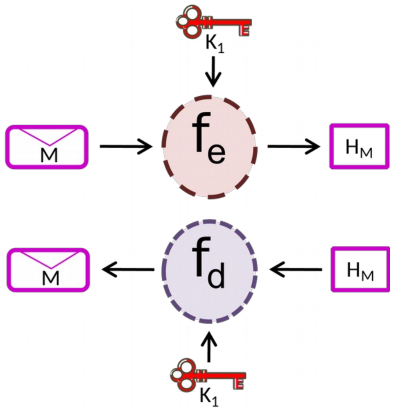
In a synchronous cryptosystem (synchronous stream cipher) the message and the keystream are generated independently of eachother. This method is generally associated with a stream cipher where the output is generated as the input arrives so the output generation of a stage is synchronous to the input arrival.

4.12 Asynchronous Cryptosystem



In an Asynchronous cryptosystem the key at each stage is based on the input to the stage and the crypto variable (key). Since the input is needed to generate the key the output of each stage must be asynchronous to the input.

4.13 Symmetric Keys



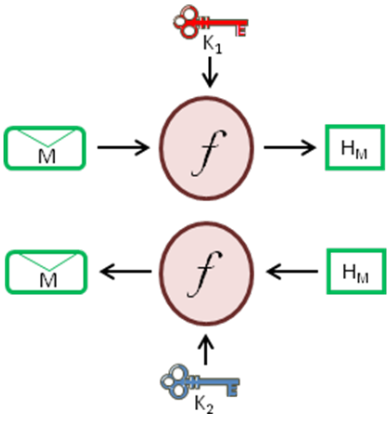
For a Symmetric system the same key is used for both encryption and decryption, therefore both sides must ensure the key is kept secret. In effect this means that the function changes and the key is contant.

4.13.1 Symmetric key limitation

The number of keys required to link n nodes are:

# of keys = [n(n-1)]/2

4.14 Asymmetric Keys



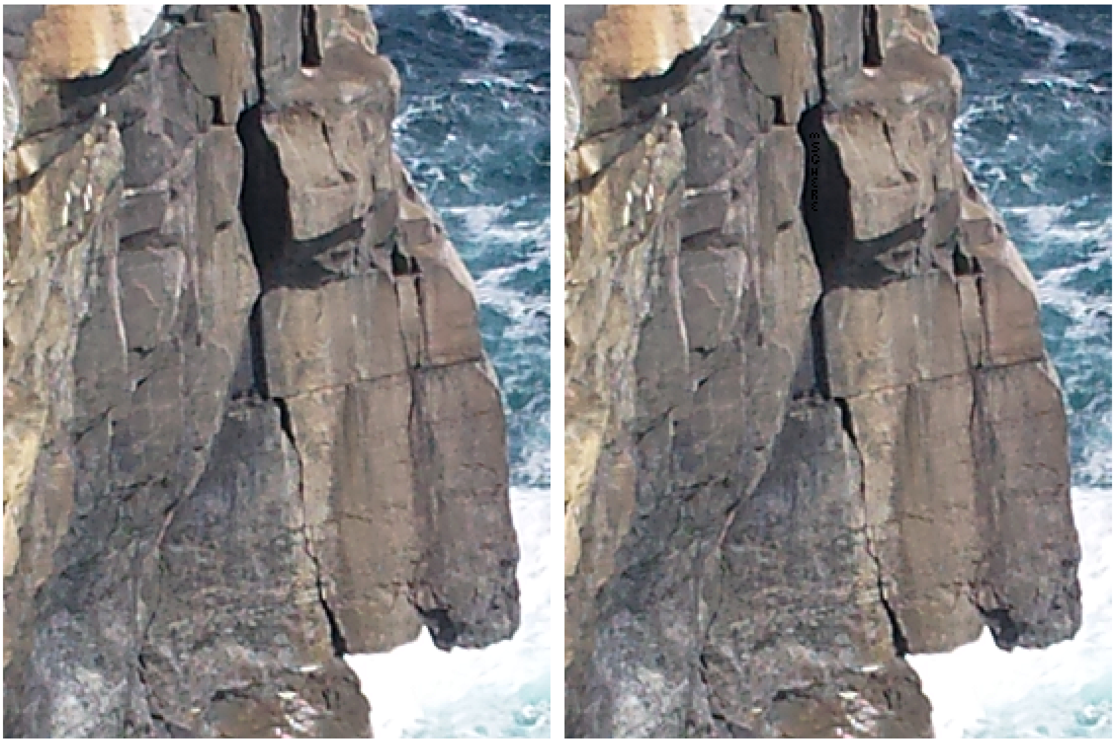
In an Asymmetric system there is a key pair, one for encryption and another for decryption. One of these keys is considered public and can be freely distributed while the second key is kept secret and is called the private key. This means that for the asymmetric system the function remains constant and the key changes.

4.15 Key Escrow

Key escrow is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorised third party may gain access to those keys. These third parties may include businesses, who may want access to employees' private communications, or governments, who may wish to be able to view the contents of encrypted communications.

4.16 Stenography

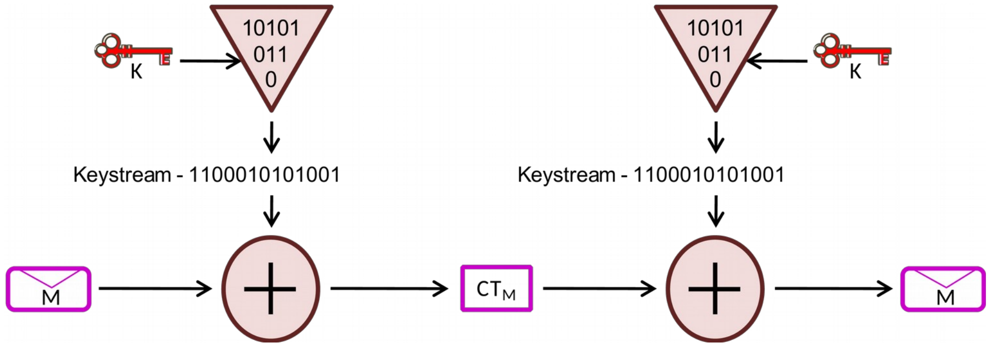
Herodotus tells us of secret messages physically concealed beneath wax on wooden tablets or as a tattoo on a slave's head concealed by regrown hair, though these are not proper examples of cryptography per se as the message, once known, is directly readable. This is known as steganography.



In digital steganography, electronic communications may include steganographic coding inside of a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. As a simple example, a sender might start with an image file and adjust the colour of every 1000th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

The pictures above give a very simplified example where a message is hidden in the shadow near the middle.

5. Stream Cipher



A stream cipher is a symmetric key cipher where plaintext bits are combined with a pseudorandom cipher bit stream (keystream), typically by an XOR function. The plaintext digits are encrypted one at a time, and the transformation of successive digits varies during the encryption. This emulates the OTP concept discussed earlier.

Stream ciphers are ideally suited to hardware implementations and are known for high speed.

5.1 Rivest Cipher 4 (RC4)

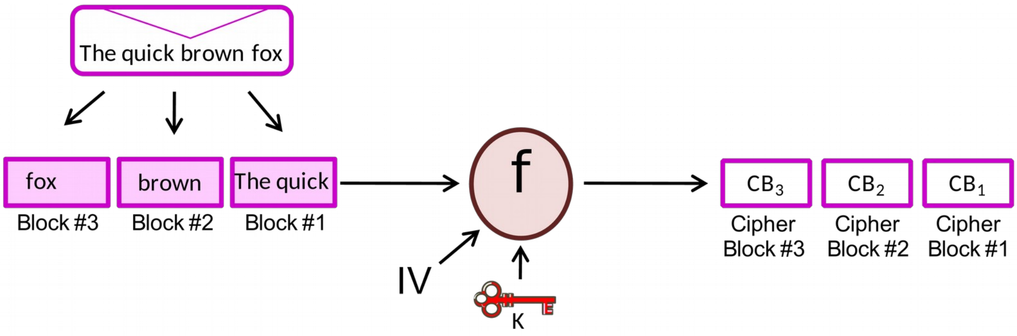
RC4 is a stream cipher designed by Ron Rivest of RSA Security, The Security Division of EMC Corporation in 1987.

RC4 is trademarked however it has become part of some commonly used encryption protocols and standards, including Wired Equivalency Protection (WEP) and Wi-Fi Protected Access (WPA) for wireless cards and Transport Layer Security (TLS).

RC4 generates a pseudorandom keystream which, is combined with the plaintext using bit-wise XOR; decryption is performed the same way. It is similar to the Vernam cipher except that pseudorandom bits, rather than random bits, are used.

It offers impressive speed and simplicity. Implementations in both software and hardware are very easy to develop.

6. Block Cipher



A block cipher is a symmetric key cipher operating on fixed-length blocks of bits. A typical block cipher algorithm could take a 128-bit block of plaintext as input, and output a corresponding 128-bit block of ciphertext. The transformation is controlled the input of a secret key.

Block ciphers can be contrasted with stream ciphers described earlier. A stream cipher operates on individual digits one at a time and the transformation varies during the encryption. The distinction between the two types is not always that clear beacuse a block cipher used in certain modes of operation acts like a stream cipher.

The initial block cipher was Data Encryption Standard (DES) from 1977 and it has been succeeded by the Advanced Encryption Standard (AES) in 2001.

6.1 Block Size

Block ciphers operate on a fixed length string of bits and this is called the block size. Both the input plaintext and output ciphertext are the same length. The output cannot be shorter than the input and it is simply undesirable for the output to be longer than the input.

Earlier block ciphers like DES used a block size of 64 bits (8 bytes). However the Birthday paradox tells us that after accumulating a number of blocks equal to the square root of the total number possible, there will be an approximately 50% chance of two or more being the same, which would start to leak information about the message contents.

AES candidates were required to support a block length of 128 bits (16 bytes). This should be acceptable for up to B = 256 Exabytes of data, and should suffice for quite a few years to come.

Note: In probability theory the ***birthday paradox*** is that in some set of randomly chosen people some pair of them will have the same birthday. In a group of at least 23 randomly chosen people, there is more than 50% probability that some pair of them will have the same birthday. Because the set is just 23 people and there is a possibility of 365 birthdays appears strange.

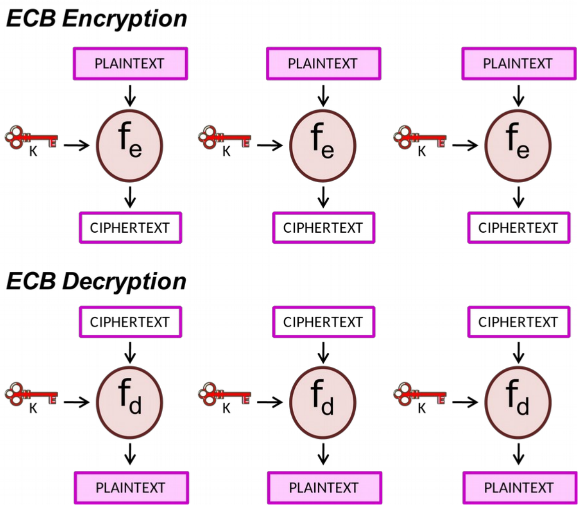
For 57 or more people, the probability is more than 99%, and it reaches 100% when the number of people reaches 366.

6.2 Rounds

Typically block ciphers are constructed by repeatedly applying a simpler function. Each iteration is called a round, and between 4 to 32 rounds are typical.

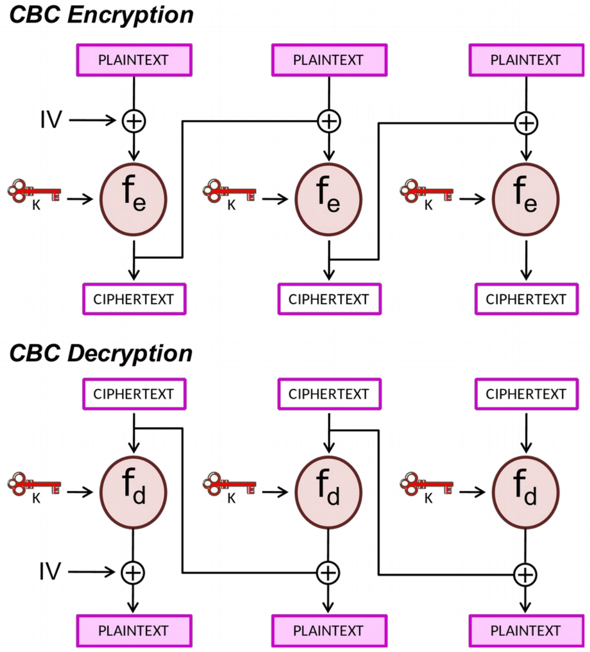
6.3 Block cipher modes of operation

6.3.1 Electronic Code Book (ECB)



The ECB is the simplest of the encryption modes. The message is divided into blocks and each block is encrypted separately. The disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks. As a result it does not hide data patterns particularly well. ECB is not recommended for use in cryptographic protocols at all.

6.3.2 Cipher-block chaining (CBC)

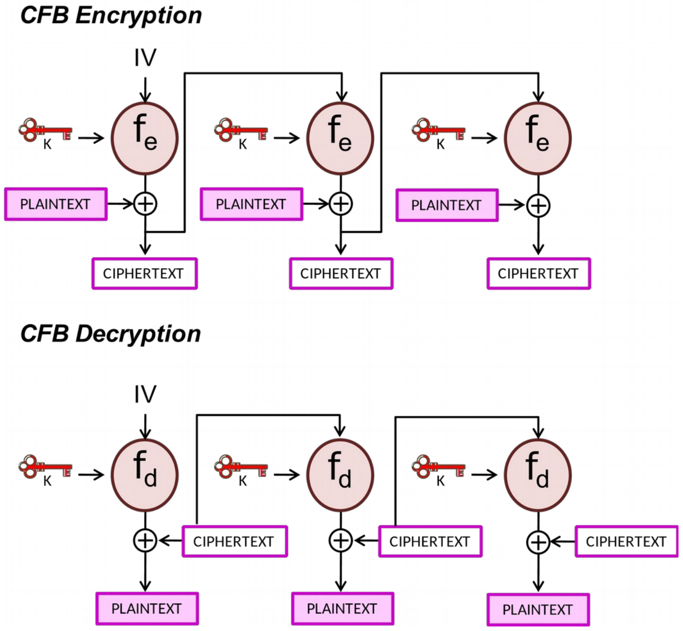


CBC Encryption

CBC Decryption

In CBC mode, each block of plaintext is XOR with the previous ciphertext block before being encrypted. This way, each ciphertext block is dependent on all plaintext blocks processed up to that point. Also, to make each message unique, an IV must be used in the first block.

6.3.3 Cipher Feed Back (CFB)



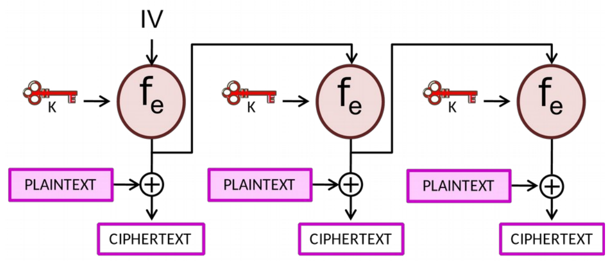
CFB Encryption

CFB Decryption

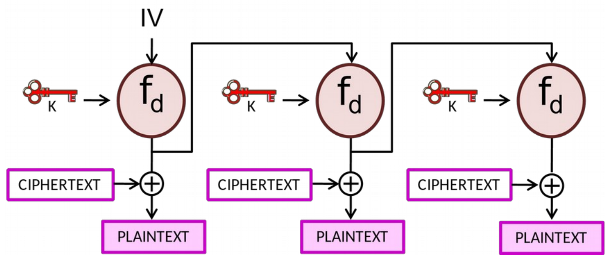
The CFB mode, a close relative of CBC and makes a block cipher into a self-synchronising stream cipher. There is very little to recommend this mode over CBC however so it sees little use.

6.3.4 Output Feed Back (OFB)

OFB Encryption



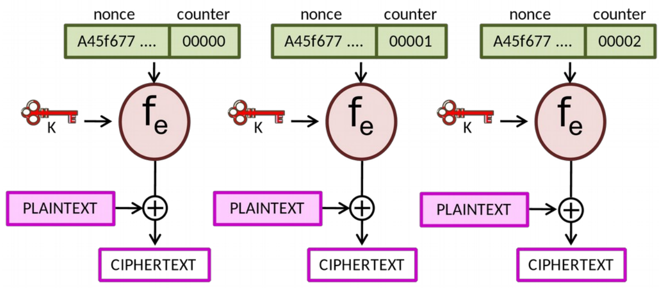
OFB Decryption



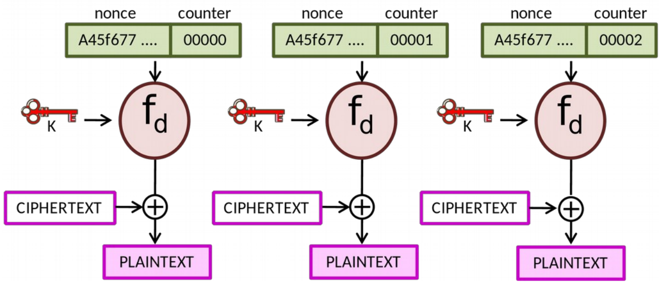
OFB mode makes a block cipher into a synchronous stream cipher. It generates keystream blocks, which are then XOR with the plaintext blocks to get the ciphertext.

6.4 Counter (CTR)

CTR Encryption



CTR Decryption



Counter mode turns a block cipher into a stream cipher. It generates the next keystream block by encrypting successive values of a "counter". The counter can be any function which produces a sequence which is guaranteed not to repeat for a long time. The nonce in the diagram is the same thing as the IV in the earlier modes. The nonce and the counter can be concatenated, added, or XOR together to produce the actual unique counter block for encryption.

6.5 Counter Mode with CBC-MAC (CCM)

CCM is a mode of operation for cryptographic block ciphers. It is an authenticated encryption algorithm designed to provide both authentication and privacy. CCM mode is only defined for block ciphers with a block length of 128 bits. CCM combines the well-known counter mode of encryption with the well-known CBC-MAC mode of authentication. The key insight is that the same encryption key can be used for both. CCM is defined for AES only in RFC 3610.

- CBC-MAC – Authentication
- Counter Mode – Confidentiality

6.6 Galois/Counter Mode (GCM)

GCM is also a mode of operation for cryptographic block ciphers with a block size of 128 bits. GCM combines the CTR mode of encryption with the Galois mode of authentication. A key feature is that the Galois field multiplication used for authentication can be easily computed in parallel thus permitting higher throughput than the authentication algorithms that use chaining modes, like CBC.

7. Data Encryption Standard (DES)

DES is a block cipher selected by NIST as an official standard for the U.S. in 1976 and published it in January 1977 as FIPS PUB 46. It went on to have widespread use internationally also. It is based on a symmetric-key algorithm that uses a 56 bit key.

DES is now considered to be insecure for many applications. This is chiefly due to the 56-bit key size being too small. It was publically broken in just over 22 hours. There are also some analytical results which demonstrate theoretical weaknesses in the cipher, although they are unfeasible to mount in practice.

7.1 Double DES (DDES)

Double DES is a version of DES which applies the algorithm twice to the plaintext with a different 56 bit key each time. It is basically encrypted with one key and re-encrypted with the second key. It does not lead to a major increase in security.

ciphertext = Encrypt K_2*(Encrypt* K_1*(plaintext))*

i.e., DDES encrypts with K_1, then DES re-encrypts with K_2.

DDES was thought to provide an effective key length of 112 bit cipher but is susceptible to a man in the middle attack.

7.2 Triple DES (TDES)

Triple DES is an enhanced version of DES which was published as an ANSIX3.92 standard and later as FIPS PUB 46-3 (1999) uses a "key bundle" which comprises three DES keys, K_1, K_2 and K_3, each of 56 bits. The encryption algorithm is:

ciphertext = Encrypt K_3*(Decrypt* K_2*(Encrypt* K_1*(plaintext)))*

i.e., DES encrypts with K_1, DES decrypts with K_2, then DES encrypts with K_3.

Decryption is the reverse:

plaintext = Decrypt K_1*(Encrypt* K_2*(Decrypt* K_3*(ciphertext)))*

i.e. decrypts with K_3, encrypts with K_2, then decrypts with K_1.

Each triple encryption encrypts one block of 64 bits of data. In each case the middle operation is the reverse of the first and last. This improves the strength of the algorithm when using keying option 2, and provides backward compatibility with DES with keying option 3.

7.2.1 Keying Options

The standards define three keying options:

- Keying option 1: (Strongest): All three keys are independent.
- Keying option 2: K_1 and K_2 are independent, and $K_3 = K_1$.
- Keying option 3: (No better than DES): All e keys are identical, i.e. $K_1 = K_2 = K_3$.

The algorithm is believed to be practically secure in the form of Triple DES, although there are theoretical attacks. In recent years, the cipher has been superseded by the Advanced Encryption Standard (AES).

8. Advanced Encryption Standard (AES)

AES was selected to replace DES, DDES and TDES as encryption standards. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each AES cipher has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively.

AES was announced by NIST as FIPS PUB 197 in November 2001 after a 5-year process in which fifteen competing designs were presented and evaluated before Rijndael was selected as the most suitable. AES also became the first publicly accessible and open cipher approved by the NSA for top secret information.

The Rijndael cipher was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, and submitted by them to the AES selection process.

9. Other Block Ciphers

9.1 Rivest Cipher 5/6 (RC5/6)

RC5 is a block cipher designed by Ronald Rivest in 1994 and patented by RSA, The Security Division of EMC Corporation. It is notable for its simplicity. It has a variable block size (32, 64 or 128 bits), key size (0 to 2040 bits) and number of rounds (0 to 255). The original suggested choices of parameters were a block size of 64 bits, a 128-bit key and 12 rounds.

RC6 is a symmetric key block cipher derived from RC5. It was designed by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin to meet the requirements of the Advanced Encryption Standard (AES) competition and was one of the five finalists. It has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits, but, like RC5, it can be parameterised to support a wide variety of word-lengths, key sizes and number of rounds.

9.2 Blowfish and Twofish

Blowfish is a block cipher designed by Bruce Schneier in 1993 as a general-purpose algorithm, intended as a replacement for DES and free of the problems and constraints associated with other algorithms. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. Blowfish is unpatented and can be used freely by anyone. Blowfish has a 64-bit block size and a variable key length from 32 up to 448 bits. It is a 16 round Feistel cipher.

Twofish is related to Blowfish and is a symmetric key block cipher with a block size of 128 bits and key sizes up to 256 bits. It was one of the five finalists of the Advanced Encryption Standard contest. On most software platforms Twofish is slightly slower than Rijndael for 128-bit keys, but somewhat faster for 256-bit keys. Twofish has also not been patented and is one of a few ciphers included in the OpenPGP standard (RFC 4880)

9.3 Secure And Fast Encryption Routine (SAFER)

SAFER is a family of block ciphers designed by James Massey of Cylink Corporation. The early SAFER K and SAFER SK designs share the same encryption function, but differ in the number of rounds and the key schedule. SAFER+ and SAFER++ versions were submitted as candidates to the AES competition. All of the SAFER algorithms are unpatented and available for open use.

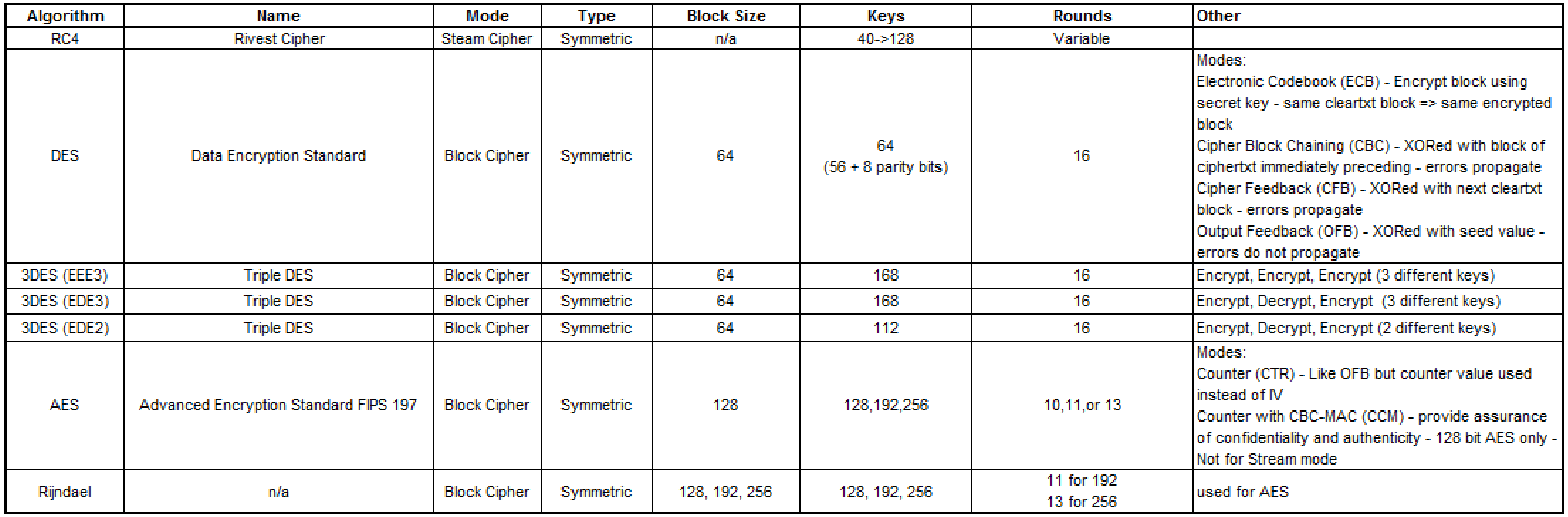
9.4 CAST

Are another family of block ciphers documented in RFC 2144 and RFC 2612. They are royalty and licence free basis for commercial and non-commercial uses. CAST-128 is a 12 or 16 round Feistel network with a 64 bit block size and a key size of between 40 to 128 bits while CAST-256 another AES candidate that was adapted for a 128 bit block size. Acceptable key sizes are 128, 160, 192, 224 or 256 bits and is composed of 48 rounds.

9.5 Serpent

Serpent is a symmetric key block cipher which was a finalist in the Advanced Encryption Standard (AES) contest, where it came second to Rijndael. Serpent was designed by Ross Anderson, Eli Biham, and Lars Knudsen. Serpent has a block size of 128 bits and supports a key size of 128, 192 or 256 bits. During design it was thought that while 16 rounds were sufficient against known types of attack, 32 rounds would be insurance against future attacks. It is a free to use unpatented public domain cipher.

9.6 Cryptographic Method Summary



Algorithm	Name	Mode	Type	Block Size	Keys	Rounds	Other
RC4	Rivest Cipher	Steam Cipher	Symmetric	n/a	40->128	Variable	
DES	Data Encryption Standard	Block Cipher	Symmetric	64	64 (56 + 8 parity bits)	16	Modes: Electronic Codebook (ECB) - Encrypt block using secret key - same cleartxt block => same encrypted block Cipher Block Chaining (CBC) - XORed with block of ciphertxt immediately preceding - errors propagate Cipher Feedback (CFB) - XORed with next cleartxt block - errors propagate Output Feedback (OFB) - XORed with seed value - errors do not propagate
3DES (EEE3)	Triple DES	Block Cipher	Symmetric	64	168	16	Encrypt, Encrypt, Encrypt (3 different keys)
3DES (EDE3)	Triple DES	Block Cipher	Symmetric	64	168	16	Encrypt, Decrypt, Encrypt (3 different keys)
3DES (EDE2)	Triple DES	Block Cipher	Symmetric	64	112	16	Encrypt, Decrypt, Encrypt (2 different keys)
AES	Advanced Encryption Standard FIPS 197	Block Cipher	Symmetric	128	128,192,256	10,11,or 13	Modes: Counter (CTR) - Like OFB but counter value used instead of IV Counter with CBC-MAC (CCM) - provide assurance of confidentiality and authenticity - 128 bit AES only - Not for Stream mode
Rijndael	n/a	Block Cipher	Symmetric	128, 192, 256	128, 192, 256	11 for 192 13 for 256	used for AES

10. Asymmetric Key Cryptography

Asymmetric key cryptography or public key cryptography is a relatively new cryptographic approach where the use of asymmetric key algorithms instead of or in addition to symmetric key algorithms is used as an enhancement to security.

Public key cryptography unlike symmetric key algorithms does not require a secure initial exchange of one or more secret keys to both sender and receiver. Instead a mathematically related key pair is created, a secret private key and a public key the latter which is published. These keys allow protection of the authenticity of a message by creating a digital signature of a message using the private key, which can be validated using the public key. It also allows for the protection of the messages confidentiality and integrity, by public key encryption, encrypting the message using the public key, which can only be decrypted using the private key.

Public key cryptography is employed by many cryptographic algorithms and cryptosystems. It is used in standards such as Transport Layer Security (TLS)/Secure Sockets Layer (SSL), Pretty Good Privacy (PGP), and GNU Privacy Guard (GnuPG).

10.1 Key pairs

The generation of key pairs requires the use of intractable problems called trapdoor functions which are functions that are easy to compute in one direction, yet believed to be difficult to compute in the opposite direction without special information, called the "trapdoor".

An intractable problem is a problem for which there is no efficient means of solving. These aren't necessarily problems for which there is no solution. Instead, these are problems that take too long to analyse all the options. The public key cryptographic intractable problems used to date are based either on factoring prime numbers or discrete logarithms.

Looking at an example:

Take a prime number $m = 29$ as the modulus (public key). The primitive roots of 29 are: 2, 3, 8, 10, 11, 14, 15, 18, 19, 21, 26, 27.

So taking and a base $b = 10$ (the trapdoor)

Alice chooses a secret $y = 8$ (Private Key).

Alice sends Bob $w = by \text{ mod m} = 10^{8} \bmod 29 = 25$

Bob chooses a secret $z = 11$ (Private Key).

Bob sends Alice $x = bz \bmod m = 10^{11} \bmod 29 = 2$

Alice computes $s = xy \bmod m = 2^{8} \bmod 29 = 24$

Bob computes $s = wz \bmod m = 25^{11} \bmod 29 = 24$

Alice and Bob now share a secret, in this case 24 without it being transferred across the transmission path and without either Alice or Bob sharing their private keys.

10.1.1 Diffie-Hellman key protocol

In 1976 Whitfield Diffie and Martin Hellman, who, influenced by Ralph Merkle's work on public-key distribution went down the discrete log route when developing what became known as Diffie-Hellman key exchange method.

10.1.2 El Gamal

El Gamal is based on Diffie-Hellman method. It was described by Taher Elgamal in 1985. It is used in the free GNU Privacy Guard software, recent versions of PGP, and other cryptosystems.

10.1.3 RSA

In 1977 Ronald Rivest, Adi Shamir and Len Adleman developed an algorithm using factoring of prime numbers. This algorithm became known as RSA.

Taking two large prime numbers we will call ‘B’ and ‘Q’. Multiply these numbers to generate ‘N’:

$$N = B * Q$$

Select another number ‘*e*’ such that:

1. $e < N$
2. e and $(N - 1)(Q – 1)$ are relatively prime (no common factors except 1)

Find a number ‘*p*’ such that:

$$(ep - 1) \bmod (B - 1)(Q - 1) = 0$$

Distribute ‘*e*’ and ‘*N*’ as the public key and keep ‘*p*’ as the private key.

For Alice to send an encrypted message she sends:

$$\{CT\} = \{PT\}^{e} \bmod N$$

Bob receives and retrieves the message by:

$$\{PT\} = \{CT\}^{p} \bmod N$$

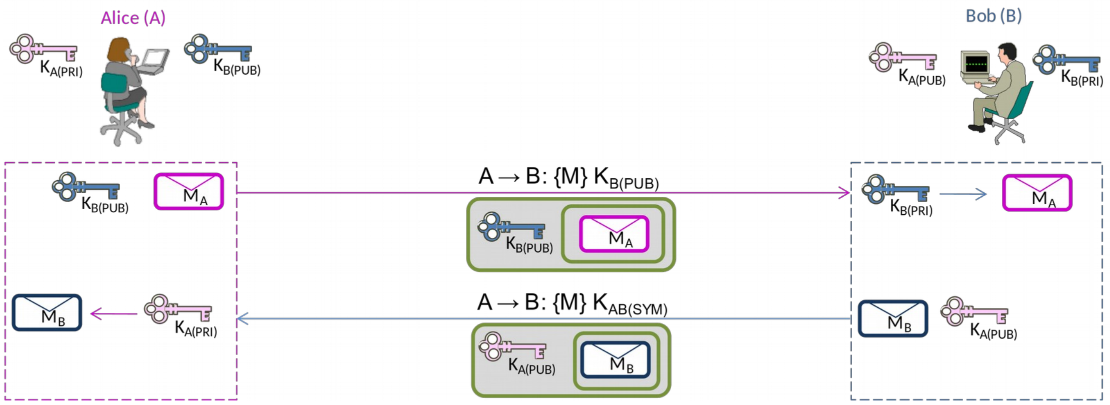
10.1.4 Elliptic curve cryptography (ECC)

Another intractable problem that is used is the assumption that finding the discrete logarithm of an elliptic curve element is infeasible. The size of the elliptic curve determines the difficulty of the problem. It is believed that a smaller group can be used to obtain the same level of security as RSA-based systems. Using a small group reduces storage and transmission requirements.

10.2 Asymmetric Key Protocol summary

Algorithm	Name	Mode	Block size	Keys	Other
RSA	Ron Rivest, Adi Shamir & Len Adleman	Factoring	Variable	1024 – 2048	
Diffie Hellamn	Whitfield Diffie & Martin Hellman	Discrete Log	Variable	Variable	Only used for key exchange
ECC	Elliptical Curve Cryptography	Discrete Log	Variable	80 → 512	160 bits key is equivalent to 1024 bits in RSA

10.3 How Asymmetric Key Cryptography works



Alice and Bob wish to communicate with each other so they each have a public key and a private key. Alice has a copy of Bob's public and Bob has a copy of Alice's public key.

Alice wishes to send a message to Bob so she encrypts it with Bob's public key $K_{B(PUB)}$ and forwards it to him. Bob extracts the message using his private key $K_{B(PRI)}$.

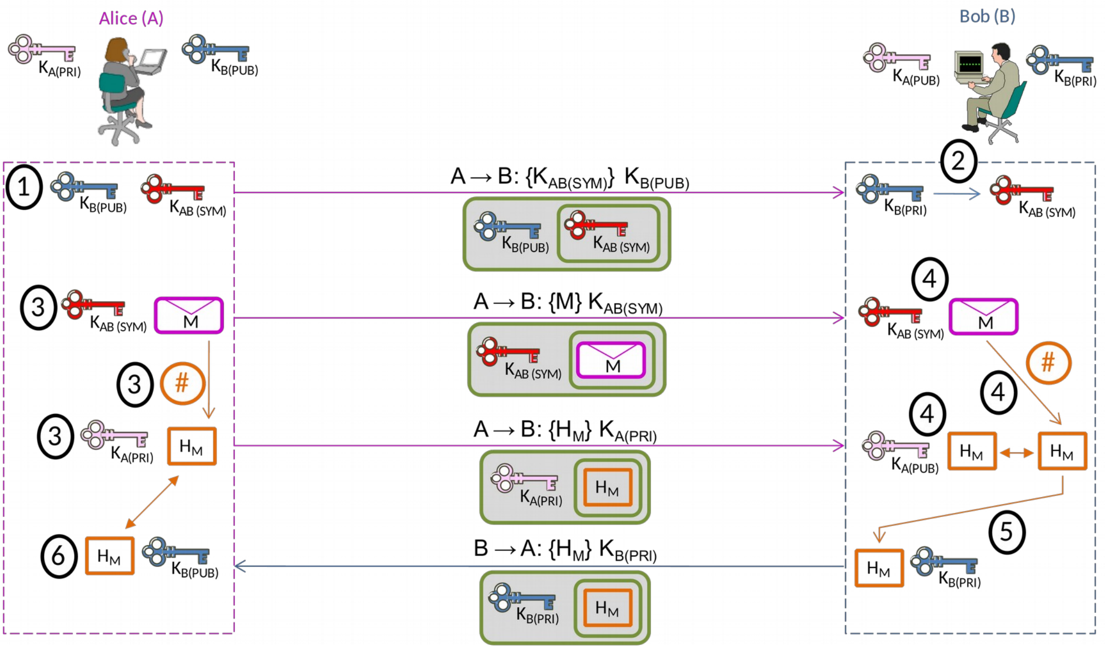
Bob wishes to respond with a message of his own to Alice so he encrypts it with Alice's public key $K_{A(PUB)}$ and forwards it to her. Alice extracts the message using her private key $K_{A(PRI)}$.

This scheme offers confidentiality of transmission from Alice to Bob and from Bob to Alice. This scheme does not however address the issues of integrity or non-repudiation.

10.4 Digital Signature

Apart from confidentiality of data another application of public-key cryptography is digital signature. Digital signature schemes can be used for sender authentication and non-repudiation. In such a scheme a user who wants to send a message computes a digital signature of this message and then sends this digital signature together with the message to the intended receiver. Digital signature schemes have the property that signatures can only be computed with the knowledge of a private key. To verify that a message has been signed by a user and has not been modified the receiver only needs to know the corresponding public key.

10.5 The hybrid system



This is an enhancement of the system on the previous page where Digital Signatures have been added to provide authentication, integrity and non-repudiation as well as confidentiality.

Alice and Bob wish to communicate with each other so they each have a public key and a private key. Alice has a copy of Bob's public and Bob has a copy of Alice's public key.

Alice wishes to send a message to Bob so she generates a symmetric key $K_{AB(SYM)}$ which she encrypts with Bob's public key $K_{B(PUB)}$ and forwards it to him. Bob extracts $K_{AB(SYM)}$ from the message using his private key $K_{B(PRI)}$.

Alice encrypts the message she wants to send using the shared symmetric key $K_{AB(SYM)}$ and forwards it to Bob, she also generates a message digest from the message and using her own private key $K_{A(PRI)}$ to encrypt the hash forwards the encrypted hash to Bob.

Bob uses the symmetric key $K_{AB(SYM)}$ to decrypt the message, this ensures the confidentiality of the message. He also generates a message digest of it, he then takes the encrypted message digest received and decrypts it using Alice's public key $K_{A(PUB)}$. He now compares the message digest received from Alice with the version he created and they should be the same. If so he is assured of the message integrity.

Finally he can acknowledge the receipt by taking the message digest and encrypting it with his private key $K_{B(PRI)}$ and forwarding it to Alice. Alice decrypts it using Bob's public key $K_{B(PUB)}$ and the output should be identical to the message digest Alice herself created. This verifies the receiver's integrity as well as assuring Alice that Bob received the message.

11. Key Management

One of the obvious issues with the asymmetric key cryptography is how to make the public keys available. For this we need a Public Key Infrastructure (PKI). This is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

11.1 Certificate Authorities (CA)

CAs are web sites that publish the key bound to a given user. This is achieved using the CA's own key, so that trust in the user key relies on one's trust in the validity of the CA's key. The mechanism that binds keys to users is called the Registration Authority (RA), which might or might not be separate from the CA. The key-user binding are established, depending on the level of assurance the binding has, by software or under human supervision.

The term trusted third party (TTP) may also be used for certificate authority (CA). Moreover, PKI is itself often used as a synonym for a CA implementation.

The ITU-T standard for Certificate Authority is included within the X.509 system.

11.2 Web of Trust

An alternative approach to the problem of public authentication of public key information is the web of trust scheme, which uses self-signed certificates and third party attestations of those certificates. PGP and GnuPG are examples of implementations of the web of trust model. They allow the use of e-mail digital signatures for self-publication of public key information; it is relatively easy to implement one's own Web of Trust.

11.3 Implementations

11.3.1 Privacy Enhanced Mail (PEM)

PEM was an early IETF proposal for securing email using public key cryptography. It has never seen wide deployment as it depended on prior deployment of a hierarchical public key infrastructure (PKI) with a single root. Deployment of such a PKI proved impossible due to cost and legal liability of the root CAs became understood. It was also seen as not a good idea to impose central authority to e-mail.

- RFC 1421 PEM: Part I: Message Encryption and Authentication Procedures
- RFC 1422 PEM: Part II: Certificate-Based Key Management
- RFC 1423 PEM: Part III: Algorithms, Modes, and Identifiers
- RFC 1424 PEM: Part IV: Key Certification and Related Services

11.3.2 Pretty Good Privacy (PGP)

PGP was created by Philip Zimmermann in 1991, it is a program that provides cryptographic privacy and authentication. PGP is used for signing, encrypting and decrypting e-mails to increase the security of e-mail communications.

PGP follows the OpenPGP standard (RFC 4880) for encrypting and decrypting data.

PGP uses a serial combination of hashing, data compression, symmetric-key cryptography, and public-key cryptography. Each public key is bound to a user name and/or an e-mail address. The first version of this system was a web of trust, however current versions of PGP encryption include both web of trust and certificate authority options through an automated key management server.

GnuPG is the GNU project's complete and free implementation of the OpenPGP standard.

11.3.3 Secure/Multipurpose Internet Mail Extensions (S/MIME)

MIME is the standard that extends the format of e-mail to support:

- Text in character sets other than ASCII
- Non-text attachments
- Message bodies with multiple parts
- Header information in non-ASCII character sets

S/MIME is a standard for adding cryptographic signature and encryption services to MIME data.

S/MIME is defined in RFC 2633. S/MIME was originally developed by RSA Data Security. However it is now managed by the IETF.

This page is intentionally blank