Cryptography for Network Security

Francisco Miguel Dios Buitrago
Åbo Akademi, 2012

***Abstract*—Cryptography has been used since antiquity in order to make texts unreadable for those who should not know anything about the message. Nowadays it is used to improve the security of network communication due to people increasingly using the Internet for shopping, for managing bank accounts or for sending any other kind of sensitive information. Cryptography allows such secure communication using solely keys. In this paper we are explaining the basics and the differences between symmetric-key and public-key cryptography, two of the most common encryption method groups in modern cryptography. Moreover, we study several algorithms such as DES and AES, two of the most popular symmetric-key methods; RSA, public-key algorithm based on factoring large integers in prime numbers; ElGamal, another public-key method whose security relies on the difficulty of computing discrete logarithms. Finally, we review some cryptographic applications that can be used to solve common issues, like authentication and digital signatures.**

***Index Terms*—Cryptography, Symmetric-key, Public-key, encryption, decryption, cipher, cryptosystem, digital signature, network security.**

I. Introduction

Cryptography has become a fundamental tool these days for keeping our privacy in any kind of peer to peer communication. The information we transmit through most of the communication media may consist of very sensitive data (for instance, confidential documents), therefore this communication is susceptible to be intercepted or altered without permission. Here is where cryptography comes in: if the data are encrypted before sending them and they are intercepted throughout the transmission it is very likely to the interceptor will not be able to use this information. Besides, if the message is altered, then the receiver might realize it when the decryption process starts. Due to this importance cryptography has been expanded and improved continuously, especially since the beginning of the computer era. Note that cryptography is not the same as stenography, which purpose is to hide the message, not to make it unreadable.

In this paper we want to give the reader a general overview about cryptography, focusing on computational cryptography and explaining how it can be categorized depending on the nature of the encryption methods it uses.

First, we define some necessary key words for understanding the concepts collected in this text:

- *Encryption/Decryption:* processes that convert clear information into non-understandable sequences and vice versa.
- *Cipher:* algorithm for performing encryption or decryption processes.
- *Cipher key:* sequence or piece of information (a parameter) that controls the operation and behavior of the cipher.
- *Plaintext:* information a sender wants to transmit to a receiver. Also known as cleartext.
- *Ciphertext:* encrypted information obtained by applying a cipher on plaintext.
- *Cryptosystem:* set of algorithms that take a cipher key as parameter and convert plaintext into cipher text and back. These algorithms usually are ciphers and one for key generation, but the last one is not always necessary. A cryptosystem is considered broken if it exists a successful attack faster than brute force search.
- *Digital certificate:* an attachment to an electronic message used for security purposes. It is provided by Certificate Authorities.

Regardless of the communication channel we would use for the information transmission (it should be as safe as possible), we are able to increase the security by using any cryptographic technique. These methods can belong to one of the two main varieties of ciphers: stream cipher or block cipher. The categorization we are speaking about is shown in Fig. 1.

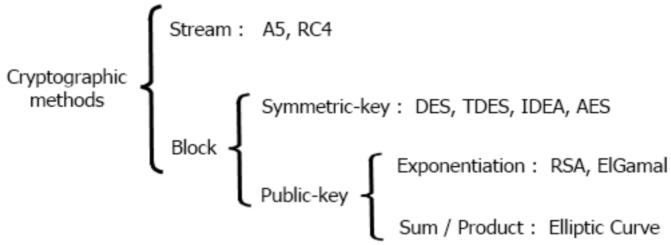

Stream : A5, RC4
Cryptographic methods
Symmetric-key : DES, TDES, IDEA, AES
Block
Exponentiation : RSA, ElGamal
Public-key
Sum / Product : Elliptic Curve

Fig. 1. Categories of cryptographic methods [3].

In general, stream cipher takes an initialization value and a secret key as input and encrypts the plaintext bit-by-bit adding a random sequence of bits (calculated using the initialization value) to the original text. Actually, the calculated random sequence is not entirely random, but satisfies certain necessary properties, such as Golomb's Postulates [1]. This kind of cipher has been widely used due to its low cost, and usually it is optimal for hardware implementation [2].

Nevertheless, in this paper we will focus on the second kind of cryptographic methods, block cipher. This method splits the plaintext into relatively large blocks (64b, 128b, 192b...) and encrypts each block separately with the same key. Block ciphers are divided into two main branches. The first one, Symmetric-key Cryptography, is based on permutations and transpositions to encrypt the text whereas the second one, Public-key Cryptography, uses complex mathematical operations for the same purpose [4].

Every secure transfer system founded on cryptography should consider these four points:

- Confidentiality
- Authenticity
- Integrity
- Non-repudiation

However, not always we can ensure these four issues using just one method, therefore, sometimes, we will have to combine many methods or applications.

We proceed as follows. In sections II and III we discuss Symmetric-key Cryptography and Public-key Cryptography, respectively. After that, in section IV we put forward different applications of block cryptography as well as cryptography in general. In section V we draw some conclusions, discussing when we should use these algorithms and why one algorithm is not always better than another.

II. SYMMETRIC-KEY CRYPTOGRAPHY

Symmetric-key Cryptography denotes a set of encryption algorithms that work as follows. First, the plain text is divided in blocks of a certain size (64b, 128b, 192b...). Then, each block is encrypted individually with the same key, thus for each block of plaintext we had we will obtain one new block of ciphertext. If the original text is not a multiple of the block size the last segment is filled until the block size is reached [3].

Generally, the encryption method of these algorithms is bidirectional, so we can obtain the original text starting from the ciphertext using the same key but doing the inverse process. This means that both sender and receiver must know the key and keep it secret.

Most of these algorithms are based on permutations and transpositions of bits at different levels. In general, the strength of this methods lies in the secret key, not in the algorithm itself [3]. This means that knowing the working process does not help attackers to break the security if they do not own the key. Actually, many of the most famous algorithms in cryptography are public, so anyone can know how each one works. The Symmetric-key Cryptography process is shown in Fig. 2.

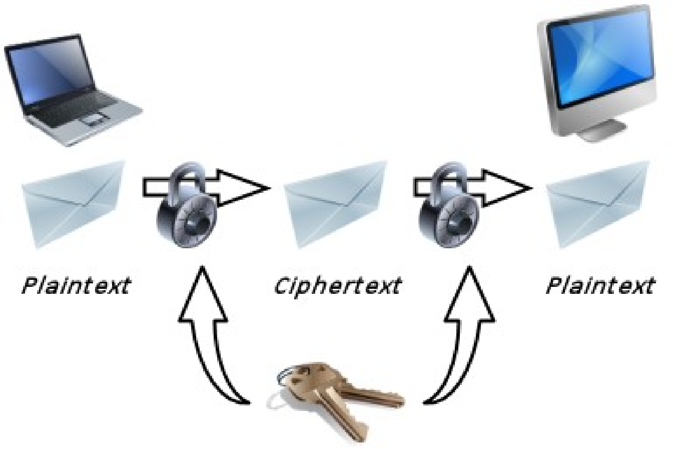


Fig. 2. Symmetric-key cryptography scheme

Two very important algorithms to highlight in this category are DES and AES. Both of them were designed in contests organized by the National Bureau of Standards in 1974 and by the National Institute of Standards and Technology in 1997 (NIST was the substitute of NBS) respectively, when they asked for proposals for a new encryption standard. We describe these algorithms in the following.

A. Data Encryption Standard

Before describing how the Data Encryption Standard (DES) algorithm works it is necessary to explain what Feistel Networks are. These were first described by Horst Feistel at IBM in order to design symmetric encryption schemes. It consists in dividing each block *i* to encrypt in two sub-blocks L_i and R_i with the same size and then apply some operations on each sub-block during a certain number of rounds where the output of one round is the input of the next one. This is illustrated in Fig. 3, inspired from [5].

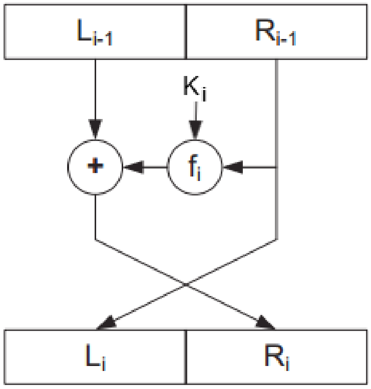


Fig. 3. One round in Feistel Network [5]

Specifically, in each round the sub-blocks are swapped and then added to the result of a function that depends on the preceding sub-block and a certain key. The last round is a bit different since there is no swapping [3].

DES works with 64-bit blocks and a 64-bit key, but actually it only uses 56 bits of this key; the rest is employed for parity checks (consequently, there are only 2^{56} possible keys).

This algorithm is based on Feistel Networks, but it adds more steps to the encryption/decryption process: at the beginning it permutes the bits inside each block in a certain way, then it applies a complex computation using the key and in the end of the algorithm it permutes the bits again in the inverse way using permutation tables. This key-dependent computation is in fact a Feistel Network with 16 rounds, using a specific cipher function *f* and 16 different keys generated from the original one (by shifting and permuting bits). This function *f* consists of permutations, transpositions and logical operations [6].

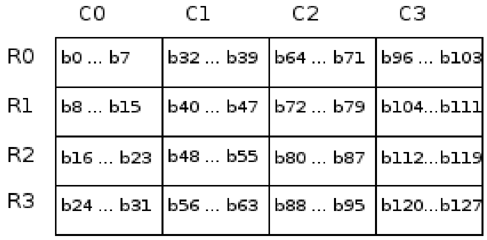
The decryption process is exactly the same, with the exception of it using the keys in the inverse order: the key which was used in the 16th round for the encryption process becomes the key used in the 1st round, and so on.

There are already some attacks that have succeeded in breaking DES. Due to this, other versions of this algorithm have been designed, such as Triple-DES [6] which applies the DES cipher algorithm three times to each data block. However, this algorithm is three times slower than DES since it requires tripling the operations, therefore other different algorithms have been devised and currently used, in order to improve efficiency and security.

B. Advanced Encryption Standard

The Advanced Encryption Standard (AES) is one of the most popular algorithms in the group of Symmetric-key Cryptography and is widely accepted as the current standard due to its high efficiency and security. AES is a fast algorithm on both hardware and software, hence it is expected to be used for long time. In fact, even the company Intel has introduced a set of instructions in their CPUs in order to offer higher performance and security using this algorithm [7].

In the beginning it was designed under the name of “Rijndael”, and it was supposed to support block sizes multiples of 32 bits with a minimum of 128 and a maximum of 256 bits. Nonetheless, when the standard was approved, the name of “Advanced Encryption Standard” was chosen and the block size was fixed to 128 bits, with a key length of 128, 192 or 256 bits [8].



	C0	C1	C2	C3
R0	b0 ... b7	b32 ... b39	b64 ... b71	b96 ... b103
R1	b8 ... b15	b40 ... b47	b72 ... b79	b104...b111
R2	b16 ... b23	b48 ... b55	b80 ... b87	b112...b119
R3	b24 ... b31	b56 ... b63	b88 ... b95	b120...b127

Fig. 4. Bits distribution table in AES

Unlike its predecessor DES, AES is not founded on Feistel Networks since it has an entirely different process. Each sub-block is organized in 4x4 matrix (see Fig. 4), with one byte per position (128 bits = 16 bytes); a set of transformations are applied to this matrix several times. There can be 10, 12 or 14 rounds, depending on the key length, and each round consists of four different transformations (except the first one that consists just of the step 4, and the last one that skips the step 3). We detail the rounds below [9]:

1. Subbytes: first it calculates the inverse [8] of each byte and then it applies an affine transformation.
2. Shiftrows: it operates on rows. The row R0 remains intact, R1 is cyclically shifted one time to the left, R2 two times and R3 three times.
3. Mixcolumns: each column (4x1 matrix) is multiplied by a given matrix 4x4, therefore the result is four 4x1 matrices or one matrix 4x4.
4. Addroundkey: one round key is added to the current state of the matrix with a XOR operation.

Similarly to the DES algorithm, AES calculates several keys (round keys) derived from the cipher key, but the process is totally different. The routine is called Key Expansion, and it generates keys for every Addroundkey phase [8][9] (the Key Expansion algorithm is independent of the process data).

The decryption is exactly the inverse of the process explained above. This means that the transformations have to be applied in the inverse order (first addroundkey, then mixcolumns, then shiftrows and finally subbytes), but also each transformation itself has to be reversed. In other words, step 1 applies inverse transformations, step 2 shifts rows to the right, step 3 multiplies in the same way but using the inverse matrix, and step 4 is exactly the same, since the XOR operation and its inverse are equal (using the same round key) [9].

There are many kinds of attacks to AES in its different versions such as boomerang attacks, described by Biryukov and Khovratovich [10], but, as the authors say, “our attacks are still mainly of theoretical interest and do not present a threat to practical applications using AES”.

The Symmetric-key cryptography, in spite of being a very powerful tool, presents several disadvantages; a very important one is that the key has to be shared between the sender and the receiver. If it is necessary to communicate over many nodes in a network, each connection has to have its own cipher key, so that the number of total necessary keys becomes very large. Furthermore, this method requires a secure channel for the key distribution, and this is not always easy to achieve. Due to these and other reasons, in the mid 70's, Diffie and Hellman present the fundamentals of Public-key cryptography [11].

III. Public-key Cryptography

The most visible difference between public-key encryption (or asymmetric encryption) and symmetric-key is that in the former we use only one key per each communication whereas in the current one we use two distinct keys per each entity. The first one is a private key, which must only be known to the owner of the key, therefore this key must not be shared with anyone else. The second key is called public-key, and it can be easily accessed by everyone (e.g. shared in a trusted remote server). This is the main advantage: it is easier to share public keys than to distribute secret keys, as in symmetric-key cryptography.

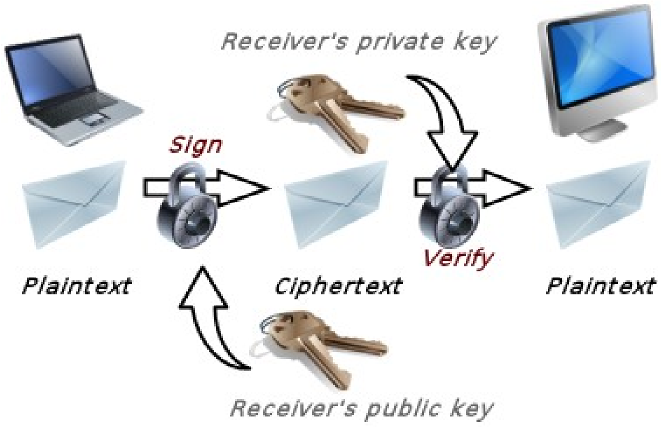


Fig. 5. Public-key cryptography scheme

The operation mode is shown in Fig. 5 and it is described as follows: the sender encrypts the information using the receiver's public-key and with a certain algorithm, then the information is sent and finally the receiver decrypts the message using his own private-key and possibly another given algorithm.

These two keys are not random: they have to be calculated by an elaborated process, and this is precisely the critical point of these kind of algorithms. It has to be computationally intractable finding out the private-key from the public-key and also the original message from the ciphertext and the public-key.

Usually, these algorithms are employed to transmit small amounts of data that are very important, therefore, the privacy and the authenticity have to be ensured. This occurs, for instance, with electronic credit card payment system. Two of the most used public-key algorithm nowadays are described below.

A. RSA algorithm

The Rivest-Shamir-Adleman (RSA) algorithm is probably the most famous and the most used algorithm among the asymmetric cipher systems. The security of RSA relies on the problem of factoring large integers as a product of prime numbers (there is no known efficient method to do this).

To calculate the two keys (public and private keys) the algorithm does the following. First, it generates two very large and distinct prime numbers p and q (random with similar bit-length). Then it calculates $n=p\cdot q$ and $\varphi(n)=(p-1)\cdot(q-1)$ (φ is called Euler's toitent function). Now, a random integer e (called encryption exponent) is selected, $1 < e < \varphi(n)$, such that $gcd(e,\varphi(n))=1$ (gcd is the greatest common divisor). Finally, it calculates an integer d, $1 < d < \varphi(n)$ so that $d\cdot e \equiv 1 \ (mod\ \varphi(n))$. The public-key is given by (n, e), whereas the private-key is d [16].

For the encryption process, the message m to be sent must be smaller than n. If it is not the case, then m has to be split into smaller pieces that n. Each piece or sub-message will be encrypted by calculating $c = m^e \ (mod\ n)$, where c is the resulting ciphertext. Moreover, a cryptogram can be decrypted by solving $m=cd \ (mod\ n)$, using the private-key [1].

We have to be especially careful when choosing the parameters of the program (i.e. prime numbers and encryption exponent), because some parameter configurations can trigger a lower security in the keys.

Many researchers have analyzed the RSA system looking for any kind of vulnerability since it was published. As Dan Boneh says in his publications about RSA [12], “Two decades of research into inverting the RSA function produced some insightful attacks, but no devastating attack has ever been found. At the moment it appears that proper implementations can be trusted to provide security in the digital world”.

B. ElGamal encryption

Like the previous algorithm, ElGamal is based on a complex mathematical operation as well. In this case, ElGamal works by calculating discrete logarithms, because this problem is also computationally unfeasible.

In order to obtain the private and public keys, the ElGamal algorithm computes the next operation: $b = \alpha^a \ (mod\ p)$, where p is a random large prime number, α is primitive element (also called generator) of $\mathbb{Z}_p$, and a is a random integer in the interval $1 < a < p-2$. The private-key is a and the public-key is given by (p, α, b).

In the encryption process, the message m has to be represented as an integer in the range $\{0,1...,p-1\}$. Then a random integer k is selected, so that $1\leq k \leq p-2$. It calculates $r = \alpha^k \ (mod\ p)$ and $s = m \cdot b^k$ (every operation is calculated $mod\ p$). The ciphertext will be the pair $(r; s)$. The owner of the private-key can decrypt this ciphertext by computing $(r^{-a}) \cdot s \ (mod\ p)$, and the result will be the original message m. More details about the key calculation or the encryption/decryption process can be found in [1][4][13].

As everything else, Public-key cryptography has several disadvantages too. The main drawback is that this kind of cryptographic methods presents a computation time that is two or three times slower than Symmetric-key methods. Hence, Public-key cryptography will not be always the right solution.

IV. Cryptographic Applications

So far we have reviewed some of the most common cryptosystems. With them we are able to make a confidential communication, so that only authorized persons could know the content; if someone outside the conversation intercepts it, then he/she would not be able to understand the information. Notwithstanding this, there are problems in network communications that require more than just confidentiality: as it was said at the beginning, secure transfers should care also about authenticity, integrity and non-repudiation. All of these points can be taken into account using little more than the techniques discussed in this paper.

In the following, we address the problem of authentication and digital signature [3]. In recent years, it has become popular to sign documents through the Internet with no need to be in person somewhere. When we sign a document, we are accepting what is written in it, and using a digital signature ensures that the document has not been altered by anyone. Otherwise, the modification would leave some proofs in the signature showing that it has been altered: signing a document guarantees its integrity. Besides, the only one who can use the digital signature of a certain person is, in fact, that person (because he/she is the only one who knows his private-key, that must not be shared with anyone), and the digital signature has registered the information of the owner. Hence, this leads to authenticity because checking the digital signature gives us the name of the owner (and possibly more information). It also leads to non-repudiation since, later, the signer cannot deny that he/she has signed the document (there are procedures to verify if it is true or not).

Remembering the working process of the public-key cryptography, we know that, if we want to send a message to another person we can encrypt the message with the receiver's public-key. Then we can send the resulting ciphertext to him/her, so that this person will be able to decrypt it using his private-key. But... what would happen if instead of encrypting a message with receiver's public-key we do with our private-key (sender's private-key)? The answer is that everyone could decrypt it using our public-key, so the confidentiality is totally lost. But if everyone can decrypt it, everyone can verify that we were the sender of that message. This is a way to achieve authenticity with the concepts described previously; concretely, this is related to public-key cryptography, RSA and ElGamal signatures. There are also several applications that use symmetric-key such as MAC (Message Authentication Code), that use the DES algorithm.

Nevertheless, it was said before that we would be able to consider authenticity, integrity and non-repudiation with little more than the techniques already described, so, what is that little more? The method mentioned above about encrypting messages with our private-key has a problem. If the message is relatively large, then the process becomes very slow (because the public-key cryptographic methods are slower than symmetric-key ones). Therefore it is infeasible to use it frequently. Another possibility would be to add a block of extra information to the original message that shows who the sender is. However, the signature ought to vary according to the associated message, because otherwise anyone could determine our signature and use it without permission, joining our signature to other messages or altering the original message without any impediment. Due to this, we are forced to generate something like a summary of the message, sign it and then join it to the original message. To achieve this we can use hash functions (also called digest functions), that basically work like that: a hash function turns any possible message into a digest of a given size, losing information in the process so that there is no possibility to recover the original text from it (it is unidirectional). This kind of functions is resistant to collisions in general, so it is not very common that two distinct messages derive in the same digest. There have been many famous hash functions, such as MD2, MD4, MD5, Tiger, FSB, GOST, but probably the most used one nowadays is SHA-1 [14].

Moreover, how can we trust that a certain private-key belongs to a certain person? In other words, if a person creates a fake digital signature or pretends to be another person, how could we be sure about the real identity? Here is where the digital certificates come in: there are several entities, such as government, police or universities that we can trust. They are committed to ensure that a person really is who he says to be since they are the responsible of the creation, maintenance or distribution of the digital certificates (this certificate can be seen as the private-key).

V. Conclusions

In this text we have shown the differences between several types of cryptographic methods in order to make the reader realize that, in general, there is no best or main method to use in every problem. We have explained that, on the one hand, Symmetric-key cryptography offers confidentiality and usually these algorithms have a high performance (in both software and hardware), but that it is very hard to keep the keys secret. On the other hand, Public-key cryptography is substantially slower than Symmetric-key cryptography but it can offer authenticity, integrity and non-repudiation if it is associated to a digital signature. Thereby they complement each other, so using them together is widespread. We can use Public-key cryptographic methods in order to distribute keys that will be used later by Symmetric-key algorithms in the encryption of the actual message to be transmitted. Thus, the security is increased without compromising the performance, simply by combining both algorithms.

References

[1] A. P. Menezes, P. van Oorschot and S. Vanstone, “Handbook of Applied Cryptography”, CRC Press, 1996.

[2] B. Preneel, "Cryptography for Network Security: Failures, Successes and Challenges," In Computer Network Security, International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, Lecture Notes in Computer Science 6258, I. V. Kotenko, and V. A. Skormin (eds.), Springer-Verlag, pp. 36-54, 2010.

[3] Dpto. Álgebra, “Notas del curso: Criptografía”, Universidad de Granada 2008.

[4] “Criptografía simétrica y asimétrica”, Universidad Politécnica de Pachuca [Online]. Available: <<http://maytics.web44.net/web_documents/criptograf_a_sim_trica_y_asim_trica.pdf >> [Last access]: 18/10/2012

[5] Prof. M. Backes, “Lecture Notes for CS-578 Cryptography, Block Ciphers”, Saarland University, 2007 [Online]. Available: <<http://www.infsec.cs.uni-saarland.de/teaching/SS07/Cryptography/ln/03-blockciphers.pdf>> [Last access]: 18/10/2012

[6] FIPS PUB 46-3, “Data Encryption Standard (DES)”, Federal Information Processing Standards Publication, 1999 October 25

[7] S. Gueron, “Intel Acvanced Encryption Standard (AES) New Instructions Set”, Intel Corporation 2012

[8] J. Daemen and V. Rijmen, “The Design of Rijndael.AES - The Advanced Encryption Standard”. Springer, 2002

[9] FIPS PUB 197, “Advanced Encryption Standard (AES)”, Federal Information Processing Standards Publication, 2001 November 26

[10] A. Biryukov and D. Khovratovich, “Related-Key Cryptanalysis of the Full AES-192 and AES-256”, Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, vol. 5912, Springer, 2009, pp. 1-18, ISBN: 978-3-642-10365-0; p. 14

[11] W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Trans. on Info. Theory, Vol. IT-22, Nov. 1976, pp. 644-654

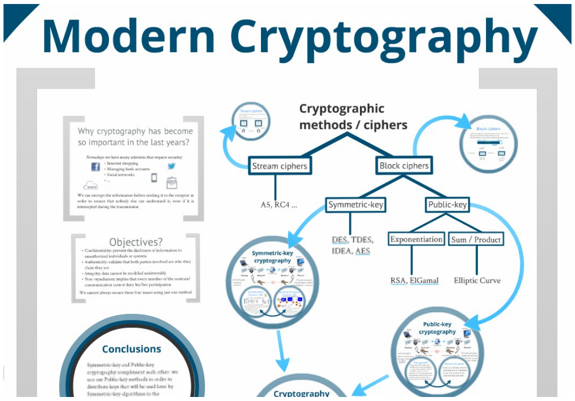
[12] D. Boneh, “Twenty years of attacks on the RSA cryptosystem”. Notices of the American Mathematical Society (AMS), Vol. 46, No. 2, pp. 203-213, 1999

[13] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms”. IEEE Transactions on Information Theory, 31 (1985), 469-472.

[14] FIPS PUB 180-2, “Secure Hash Standard”, Federal Information Processing Standards Publication, 2002 August 1

Additional material

A dynamic presentation has been designed in order to explain the content of this paper during class.



Modern Cryptography

Why cryptography has become so important in the last years?

Nowadays we have many activities that require security:
- Internet shopping
- Managing bank accounts
- Social networks
- ...

We can encrypt the information before sending it to the receptor in order to ensure that nobody else can understand it, even if it is intercepted during the transmission

Objectives?
- Confidentiality: prevent the disclosure of information to unauthorized individuals or systems
- Authenticity: validate that both parties involved are who they claim they are
- Integrity: data cannot be modified undetectably
- Non-repudiation: implies that every member of the contract/communication cannot deny his/her participation

We cannot always ensure these four issues using just one method

Conclusions

Symmetric-key and Public-key cryptography complement each other: we can use Public-key methods in order to distribute keys that will be used later by Symmetric-key algorithms in the

It shows the main concepts explained above and some additional information, such as a brief history and examples of cryptography. This presentation may help to understand the concepts mentioned in this paper.

This material can be found in: <<http://http://prezi.com/d-pzplyj1-gy/cryptography-for-network-security/>>