EVALUATION OF CRYPTOGRAPHIC ALGORITHMS

Mircea Andraşiu[1]
Emil Simion[2]

Abstract

This article represents a synthesis of the evaluation methods for cryptographic algorithms and of their efficiency within practical applications. It approaches also the main operations carried out in cryptanalysis and the main categories and methods of attack in order to clarify the differences between evaluation concept and crypto algorithm cracking.

Keywords: cryptology, cryptanalysis, evaluation and cracking cryptographic algorithms.

1. Introduction

Cryptology is the science of secrecy writings and its goal is the protection of data and information confidentiality with cryptosystems support.

Cryptography is the defensive part of cryptology, its activity field being the design of cryptosystems and of used rules. People doing this job are called *cryptographs*.

Cryptanalysis is the offensive part of cryptology, its activity field being the analysis of its own cryptosystems in order to get them the proper characteristics so they accomplish the functions intended for. Cryptanalysis can also analyze cryptosystems of third parties through their cryptograms support. The specialists in this field are called *cryptanalysts*, or, using a more romantic word, *codes crackers*.

Cryptographic algorithm means a lot of reversible transformations through which the large amount of plain texts is transformed in the large amount *M* of cryptograms.

The encryption key is a particular convention such as a word, phrase, number, numeric stream etc. that defines the encryption rule.

Cryptographic protocol is a set of rules, between two or more parties, permitting an authentication operation and/or a key or a message exchange.

A cryptosystem is composed by three elements: a cryptographic algorithm, a keys generation system and a protocol for keys distribution.

[1] University of Wales, Basarabia Bul. no. 47, sector 2, Bucharest, Romania, e-mail: mircea_andrasiu@yahoo.com
[2] University of Bucharest, Academiei 14th street, Sector 1, C.P. 010014, Bucharest, Romania, e-mail: esimion@fmi.unibuc.ro

Over ciphering consists of a lot of transformations made on cryptograms and it has the role of strengthens the cryptograms resistance (and the strength of cryptosystem, too) against the attacks of third parties cryptanalysts.

Deciphering is the opposite operation of enciphering and represents the application of the known ciphering system (in the presence of the right key) over cryptograms in order to discover the plain text.

Decryption is the operation that allow, based on the analyses of cryptograms realized with an unknown cipher system, to reveal the plain text that has been encrypted and to determine the characteristics of the cryptosystem used for ciphering.

Cryptosystems (ciphers, codes or a combination of them) are applied on plain texts edited in some language having structural and statistic characteristics according to that language. By applying cryptosystems, these characteristics are disturbed, the intensity and direction of these disturbances being found in cryptogram. A cryptosystem is better the more is the intensity of disturbances, so the cryptogram is not able anymore to reflect the structural characteristics of plain text. Different techniques and methods allow that some kind of systems to be invariant to some parameters. These invariants form the basic elements in cryptanalysis (see Tilborg [12] and Schneier [10] for an introduction in the cryptology field).

We shall exemplify, in section 2 the main operations, which are done in cryptanalysis, the starting point being the design principles of cryptographic algorithms, evaluation criteria, operations that are performed in the cryptanalysis activity. Also we present the programming problem of cracking of a cipher like the dual of the programming problem of the evaluation.

Section 3 presents a taxonomy of the main cryptanalytic attacks (attacks on encryption algorithm, attacks against the keys, attacks against authentication protocols and side channels attacks.

2. Operation in cryptanalysis

2.1. Cryptanalytic principles

Generally, designing a cryptosystem requires the meeting of the following *design principles*:

1. Don’t sub-estimate the adversary;
2. Only a cryptanalyst can assess the security of a cryptosystem;
3. For assessing a cryptosystem it has to be taking into account the fact that the adversary has complete knowledge about the evaluated system (Shannon [9]). The entire secret of a cryptosystem has to rely only on the secret key;
4. Superficial complications can be illusory and these can induce in cryptographist mind a feeling of false security;
5. All elements have to be taken into account, such as regulation regarding keys distribution.

2.2. Evaluation criteria

Claude Shannon has proposed the following elements to be taken into account when we analyze a cryptosystem:

1. The adversary's gain from the possible material decryption;
2. The length of key and the complexity of key management;
3. The complexity of a cipher-decipher cycle;
4. The size of ciphered text related to the size of plain text;
5. The way of errors propagation.

2.3. Four main operations of cryptanalysis

Usually, the main operations for resolving any cryptogram are synthesized in the following steps:

1. Establishing the language used in the plain text;
2. Establishing the type of the cryptosystem used;
3. Reconstruction of a specific key (partial or incomplete) of the equivalent cryptosystem established at step 2;
4. Reconstruction of using such a system and/or establishing complete plain text.

In some cases, step 2 can precede step 1. This is the traditional approach in cryptanalysis and can be summarized:

1. Data arranging or rearranging in order to find non-random characteristics or manifestations (frequency numbering, redundancy, forms, symmetric phenomena etc.);
2. Recognize non-random characteristics or manifestations when they are stand out in relief (through statistics or other techniques);
3. Explain the non-random characteristics or manifestations when they are recognized (by chance, brightness or perseverance). The hardest work is to establish the general structure of the system.

In the final analyses, the solution of every cryptogram involves a type of substitution that depend on reducing mono-alphabetic terms if the ciphered text is not expressed by plain text letters.

2.4. Evaluation and cracking

Evaluation is a process intended for highlight some unconformities or deficiencies of a cryptosystem which can be used by a cracker.

The evaluation of a cryptographic module can by done using NIST FIPS 140-2 standard (structured on fourth levels) and the evaluation of a product can be done using Common Criteria (ISO 15408), methodology adopted by USA, Canada and EU (structured on seventh levels).

Cracking represents an operation helping to design a technique, method or algorithm that permit the recovery of the system key or of the plain text having a reduced complexity than brute force attack method:
- the *evaluator* wants to find the *minimum quantity* of output information that help him to determine, using some strong mathematical tools, a series of information about the cipher algorithm, used key and/or plain text;
- the *cracker* wants to find the *maximum quantity* of information that help him to deduce the plain text.

The terms minimum and maximum have a general meaning. In fact, there is a problem about multicriteria decision (see Preda [7] for an introduction into Statistics Decision Theory): a series of objective functions have to be maximized (the size of ciphering key, the grade of nonlinearity, the complexity of equivalent linear, the period of pseudorandom generator, the risk of key interception in a crypto network etc.), and other function have to be minimized (key generator redundancy, the grade of correlation of inputs/outputs). These functions are related to the cryptosystem (the adversary has complete knowledge about the cryptosystem used), so the strengths of a ciphering system have to rely only on the secrecy of the key.

We take a note $Info_n(\mathbf{m},\mathbf{c},\mathbf{k})$ for the additional information regarding the cryptosystem, so a vector relation between a collection of n plain texts, a collection of $p(n)$ cryptograms and a collection of $q(n)$ particular keys. The relation $Info_n(\mathbf{m},\mathbf{c},\mathbf{k})$ is built with the support of more types of attacks, such as: attack based on plain text (known plain text and/or chosen plain text), differential attack, knowing a collection of particular keys ($q(n)>0$), identical texts ciphered with two or more different keys (to a plain text correspond more ciphered texts).

For evaluator. Let us consider $e_n \in [0;1]$ a sequence of real numbers and objective function n-p(n)-q(n).

The optimization problem for evaluator is:

$$\begin{cases} \min (n + p(n) + q(n)) \\ \mathrm{Info}_n(m, c) = 0 \\ H(m \mid c) \geq e_n, \end{cases} \qquad (1)$$

where H(**m**|**c**) is conditioned entropy (see Guiaşu [4] for the definition of conditional entropy) of the vector of **m** plain texts by the vector of cryptograms **c**.

The evaluator wants that:
1. $\lim_{n\to\infty} e_n = 1$ (knowing ciphered text doesn't compromise plain text);
2. to minimize the maximum loss (noted here by L(x, y)), thus:

$$\alpha = \min_x \max_y L(x, y),$$

where x is the *defense strategy of the evaluator* (called defense policy too), and y is attacker strategy.

For cracker. Let us consider $d_n \in [0;1]$ a sequence of real numbers and objective function $n+p(n)+q(n)$.
The optimization problem for cracker is:

$$\begin{cases} \max(n + p(n) + q(n)) \\ \text{Info}_n(m, c) = 0 \\ H(m \mid c) \le d_n \end{cases} \qquad (2)$$

where H(**m**|**c**) is conditioned entropy of the plain texts by the vector of cryptograms **c**.

The cracker wants that:
1. $\lim_{n\to\infty} d_n = 0$ (knowing ciphered text doesn't compromise plain text);
2. to maximize the minimum gain (noted here by L(*x*, *y*), the gain being a negative loss), thus:

$$\beta = \max_y \min_x L(x, y),$$

where *x* and *y* have the same specifications.

In general we have $\beta \le \alpha$, the maximum of minimum gain of the cracker can't oversize the minimum of maximum loss of the evaluator (if we have equality it means the appropriated strategies are called saddle points for *L* function.

We have $\beta < \alpha$, if interception channel is with perturbation.

The two programming problems (evaluator/cracker) are *dual problems*.

We have the following vector relation:

$\mathbf{c} = f(\mathbf{m};\mathbf{k}_t)$,
where *f* is the ciphering operator.

If $\mathbf{k}_t = \mathbf{k}$ for every $t \in T$ (*T* is the ciphering period which is a discreet collection) then the above relation is rewritten:
$\mathbf{c} = f(\mathbf{m};\mathbf{k})$, where *f* is the enciphering operator. In this case we say there is about a codification of the information (the role of codes theory is to protect information against error that can appear on the communication channel; the role of the cryptography is to protect the information against unauthorized interception).

In case of codification after resolving an nonlinear system, we can write:

$\mathbf{m} = h(\mathbf{c};\mathbf{k})$. (3)

So knowing *f* allow us to determine **m** from **c**. System (1), which is a stochastic system is more difficult to resolve then system (2), which is a determinist system, because it doesn't have the *t* parameter. The solution of system (2), given by (3), is a particular solution of system (1) for the case $\mathbf{k}_t$ being equal to **k**. In another words, we can say that encoding operation is an operation of ciphering with a particular key.

Many times, the function of ciphering *f* is given is scalar form:
$\boldsymbol{c}_i = f(\boldsymbol{m}_i,\boldsymbol{k}_i)$, for every $i=1,\ldots,n$,
where $\boldsymbol{k}_i$ is the key obtained from secret key $\boldsymbol{k}_t$.

If f can be factorized like this:

$f(m_i, k_i) = m_i \oplus g(k_i)$, where $\oplus$ is summing operator mod 2, than the encryption scheme is called *stream ciphering* and function *g pseudorandom number generator*. Because of the simplicity (from the point of view of implementation) this scheme is used in ciphering data and voice communications. In this case, the difficulty of cracking is equivalent with the difficulty of prediction or even of determination of *g* function. The technique of solving the problem is equivalent with the technique of reverse engineering.

If we want to design a good cipher we have to guarantee a minimum complexity of cracking of $O(2^n)$ (this means the opponent can't crack the system in polynomial time or, with another words, the more efficient cracking method is exhaustive searching for key or password), and if we want to design a cracker procedure for a ciphering algorithm then we have to guarantee a complexity of at least O(n) (this means that we crack the adversary system in a time no more then polynomial).

The evaluation of the cryptosystems complexity is part of confirmation tests. These tests are made, usually, after reference test processing (statistical tests or other functional criteria: strict avalanche, balance, no linearity, symmetry, no degeneration, no correlation. Tests for ciphering system evaluation are made in this order:

STEP 1. Performs references tests: *statistical tests* (see NIST Special Publication 800-22 [14] for a statistical test suite). If the ciphering system fall these tests (multicriteria decision) than it reject this cipher system, and, in opposite case, it performs STEP 2. These tests are processed in O(1) time with O(1) memory cost and they have a sensitivity usually bigger than 95%.

STEP 2. Performs references tests: *functional tests* (see Simion [8] for the definition of functional tests). If the ciphering system fall these tests (multicriteria decision) than it reject this cipher system, and, in opposite case, it performs STEP 3. These tests are processed in O(*n*) time, O(1) memory cost and they have a sensitivity usually bigger than 98%.

STEP 3. Performs *confirmation tests*: linear complexity tests. If the ciphering system fall these tests (multicriteria decision) than it reject this cipher system, and, in opposite case, it performs STEP 4. These tests are processed in O(*n*) time with O(*n*) memory cost and they have a sensitivity usually bigger than 99%.

STEP 4. Performs *confirmation tests*: Lempel-Ziv tests and squared complexity tests. If the ciphering system fall these tests (multicriteria decision) than this cipher system has a hidden predictability, and, in opposite case, it pass the tests suite. These tests are processed in $O(2^n)$ time with $O(2^n)$ memory cost and they have a sensitivity usually bigger than 99.9%.

It is obvious the faster tests are those from steps 1 and 2 and the slowest tests are from step 4.

3. Classifications of cryptanalytic attacks

This paragraph presents a series of attacks against cipher systems. There are attacks against cipher algorithms, keys, authentication protocols, system itself and unconventionally attacks (side channel attacks). These types of attacks are not exhaustive, an efficient attack being composed, usually, from a sub-collection of the followings:

3.1 Types of attacks against cipher algorithms

The main types of attacks related to the cipher algorithms are the following:

Attack with known plain text. The cryptanalyst got a ciphered text and its correspondent in plain. By this mean, the cryptanalyst intend to separate the text information from the cipher key, having the possibility to obtain, by specific methods, the cipher algorithm or a part of it and/or the key.

Attack with chosen plain text. The cryptanalyst can indicate the plain text that is to be encrypted. By this mean, the cryptanalyst intend to separate the text information from the cipher key, having the possibility to obtain, by specific methods, the cipher algorithm and/or the key.

Attack with ciphered-ciphered text. The cryptanalyst got a plain text and its correspondent text encrypted with two or more different keys. By specific methods, the cryptanalyst can restore the cipher algorithm or a part of it.

Divide and conquer attack. The cryptanalyst can realize a series of correlations between different incomings (inputs) in algorithm and its outgoing (output), trying to separate different incomings (inputs) in algorithm, this helping him to divide the problem in two ore more problems easier to resolve.

Linear syndrome attack. This method consist in the elaboration of a linear equations system of pseudorandom generator and the verification of these by the ciphered text, obtaining the plain text with a high probability.

Linear consistency attack. This method consist in the elaboration of a linear equations system of pseudorandom generator starting from an equivalent cipher key and the verification of the system by the pseudorandom generator with a probability close to 1, obtaining the plain text with a great probability.

Stochastic attack against generator outgoing (output), also called *attack by prevision* (forecast), is possible if the outgoing of the generator is correlated, the cryptanalyst succeeding to get, as input data, the output of pseudorandom generator and ciphered text, obtaining in this way the appropriate plain text. In order to avoid this type of attack, the generator must meet the following requirements:

- balance: all possible inputs have to produce all possible outputs for the same number of times;
- non-degeneration: output depends of all the elements of the input;
- immunity at correlation: correlated inputs generate uncorrelated outputs;
- strict avalanche: the change of a bit at input has to produce changes of 50% at output.

Linear informational attack against generator, also called *linear complexity attack*, is possible if the generator can be amounted to an algorithm type Fibonacci, and if the

equivalent linear complexity of the generator is small. With this techniques support it is possible to build an equivalent algorithm and an equivalent cipher key.

Attack with the period of the pseudorandom generator support is possible if the period of the pseudorandom generator is small and can rebuild the appropriate plain text.

Attack with IT viruses support is possible if the cipher algorithm is implemented and run on vulnerable or unprotected PC. The virus can substitute or inhibit the cipher algorithm being used.

3.2 Types of attacks against keys

These are the most often attacks against cipher keys:

The brute force attack consists in exhaustive checking of keys or passwords and it is possible if:

- the length of key or passwords is short;
- the space of key or passwords is small.

The smart brute force attack can be realized if the degree of key or password randomness is small (small entropy) and allow finding passwords similar with words from the language being used.

The backtracking attack consists in implementation of the method of looking type backtracking (that assume the existence of conditions for continue searching in the proper direction, see Knuth [6]).

The greedy attack provides optimum local key, which can be, or not the same as global optimum key.

The dictionary attack (searching passwords or keys is done using a dictionary) is possible if password or key are words having sense (names, data etc.)

The hybrid dictionary attack is possible by modifying words from dictionary, initializing the brut attack with dictionary's words support.

The attack with IT viruses support is possible if the keys are stored on an unprotected PC.

The attack against the hash of the password/key is possible if hash is short or inappropriate elaborated.

The substitution attack is performed when a third person substitute the original key and replaced in entire network (or a part of it). It is possible with IT viruses' support.

Storing the cipher key in an inappropriate way (together with encrypted data), in plain, without measures of physical or cryptographic protection (software or hardware), can lead to an attack against encrypted message.

Improper storing of old keys can lead to compromise old encrypted documents.

Key compromise. If the asymmetric key is compromised, only those document encrypted with this key are compromised too. If the public key is compromised, and the key may be stored on different servers, the attacker can substitute the real user causing damages in the entire communications network. Thus, the existence of master keys or of backup keys represents breaches in cryptosystem.

Conclusions: the existence of a system for key generation and management is a *sine qua non condition* in order to minimize the probability of a succeeded attack against cryptographic keys.

3.3 Types of attacks against authentication protocols

The authentication protocols are the subject to the following types of attacks:

Cryptographic attack against the public key used for signature within protocol (if the public key infrastructure is used).
Cryptographic attack against symmetric algorithm used for signature within an authentication protocol (is the symmetric system is used).
In order to avoid *the attack against digital signatures* the signature must be accomplish the followings requirements:

- the signature can not be tampered: the signature is a proof that the issuer has signed deliberately the document;
- the signature is authentic: the signature persuade the recipient that the issuer has signed deliberately the document;
- the signature is not reusable: the signature is part of the document and it can't be moved on another document;
- the signed document can be altered: after the signature process, the document can't be changed without detection;
- the signature is non-repudiated: the issuer can't pretend later that he hasn't sign the respective document.

There are some types of special digital signature like: *the invisible signature* that can be read only by the recipient of the document and the *fail-stop signature* that is a cryptographic protocol when the issuer can bring proofs if his signature has been changed.
Birthday attack, is possible if there is a high probability that signature applied on two different documents to produce the same signature.
Passive attack against authentication protocol. The interceptor monitors the communication on channel without doing any intervention, his goal being to produce conclusions about the authentication process.
Attack through a third person. The communication between the two partners of the communication channel is active intercepted by a third party.

3.4 Types of attacks against the system

The cipher system (algorithm, key and authentication protocol) can be the target to the following types of attacks:

Attack at algorithm level. These types are mentioned above.

Improper use of cipher algorithm:

- there is no message key algorithm;
- improper use of over encryption can lead to an equivalent algorithm which is weaker than every individual algorithm.

Attack at key level. These types are mentioned above.

Attack on authentication protocol or on key distribution protocols.

Attacks generated by implementation errors.

3.5 Hardware attacks against cryptographic modules

The following methods of attacks require a series of hardware measurements on crypto module:

Timing attacks. By measuring the time required to do some operations against private key, the attacker can determine the exponents used in Diffie - Hellman protocol, RSA factor, as well as a series of other crypto systems such as digital signature algorithm – DSS (see Kocher [5]).

Simple power analysis. The attack with the support of simple power analyses (SPA) consist in measuring the power consumed by device during crypto operation. This kind of attack is applied, usually, to devices with extern source of voltage (as smart-cards). The power demand depends on the instruction executed. Thus, by monitoring the power demand, we can deduce the sequence of instructions (the source code). If the sequence of instructions depends on the key length, than the power demand can give information about the key. In most processors, the pattern of the power demand by an instruction depends on the operators values (as an example, setting a bit into a registry require more power then deleting it). Measurements on more incomings can deduce the operator value. This technique is called Differential Power Analysis (DPA).

Attacks with hardware errors support. Hardware equipments can generate errors (transient, latent or induced) during arithmetic operations. By rationale exploitations of these errors it is possible to recover the private key for RSA and Rabin signature algorithms. Other cryptographic protocols such as Fiat - Schamir and Schnorr can be broken using the results of these errors (see Boneh [3]).

Differential fault analysis. Differential fault analysis (DFA) is a scheme used to recover the secret keys of a crypto system from a physical protected HSM device (Hardware Security Module). The model (see Biham and Shamir [2]) is that of random failure or induced failure. This method helps for keys identification in case of using known ciphers (as DES) and/or unknown algorithm cipher (as SkipJack).

Conclusions

This article approaches *a status quo* of analytical results regarding the typology and categorization of all types of attacks met in nowadays cryptology. It has to be taken into account that behind every class and type of attacks there are standards, mostly IT and INFOSEC (CRYPTO), and also strong mathematical and technological supports. On the

other hand, from the crypto experts' point of view, it has to be stated that the above categorization is not so incomprehensible, so the success in cryptanalysis requires many times complex attacks of more types and the existence of additionally information worth more than a standardized attack. Comparative analysis of these types of attacks would be very useful, stating that even this research doesn't get theoretical news in the field this approach can be extremely useful for day to day practice.

References

[1] Andraşiu M., *Current approaches in modern cryptology*, Journal of Information Systems and Operations Management, vol. 4, no. 1, 2010.
[2] Biham E., Shamir A, *Differential Fault Analysis of Secret Key Cryptosystem*, CRYPTO '97 Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, 1997.
[3] Boneh D., *On the importance of Checking Cryptographic Protocols for Faults*, Journal of Cryptology, Springer-Verlag, Vol. 14, No. 2, pp. 101-119, 2001.
[4] Guiaşu S., *Information theory with Applications*, McGrow-Hill, 1977
[5] Kocher P., *Timing attacks on implementation of Diffie-Hellman, RSA, DSS and other systems*, CRYPTO '96 Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, 1996.
[6] Knuth D., *The art of IT programming*, Teora Publishing House, 1999.
[7] Preda V., *Statistic Decision Theory*, Romanian Academy Publishing House, 1991.
[8] Simion E., Preda V., Popescu A., *Cryptanalysis. Mathematical Techniques and Methods.*, University of Bucharest Publishing House, 2004.
[9] Shannon C. E., *Communication Theory of Secrecy Systems*, Bell Systems Technical Journal, vol. 28(4), page 656–715, 1949.
[10] Schneier B., *Applied cryptography with source code in C*, Adison-Wesley, 1998.
[11] Stallings W., *Cryptography and network Security: Principles and Practice*, Prentice Hall, Second edition, 1999.
[12] Tilborg, H. C. A. Van, *Fundamentals of Cryptology*, Kluwer Academic Publisher, second edition, 2001.
[13] *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, NIST Special Publication 800-22, 2000.