7th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
Suceava, Romania, May 27 – 29, 2004

ENCRYPTING OPTIMISATION TECHNIQUES WITH PARTIAL AUTHENTICATION

Cătălin CERBULESCU[1], Monica CERBULESCU[2]
[1] *Faculty of Automation, Computer and Electronics, Craiova*
[2] *Carol I College, Craiova*
[1] *ccerbulescu@nt.comp-craiova.ro,* [2] *mcerbulescu@hotmail.com*

Abstract*. In order to obtain an efficient encryption, in practice it is recommended to develop schemas that use both symmetric and asymmetric key encryption. So we can benefit from the advantages of both, such as: speed and safety. The main encrypted text it is first encrypted with a session symmetric key and that key, at his turn it is encrypted with a public key. For the stream encrypted with the public key there are authentication algorithms while there are not for symmetric encryption. The present solutions for the problem solve the message authentication through a hash function. This approach has two disadvantages: high computational times and message length increase. This paper proposes a simple but intuitive solution for solving the problem in situation in witch the probability that the message will be affected with some types of modification it is low enough.*

Keywords: *symmetric key encryption, public and private keys, message authentication, message length, error detection, probability that the message will be affected.*

Introduction

A very common used in practice encryption technique consists in combining, using several algorithms types, of the symmetric and asymmetric key encryption. So, it's trying to obtain fast but safe encryption schemas. The safety in this case means that the enemy will not decrypt the message in useful time rather than he can't decrypt them.

The common attacks on encrypted messages where, initially, concentrated on decrypting messages, without the necessary knowledge of the encryption key. This was not possible once the more and more sophisticated encryption schemas were used. Once those schemas were used, for both symmetric and asymmetric key encryption, the attacks were concentrated on a exhaustive search of the decryption key.

A necessary but not enough condition that an encryption schema will be safe is that the decryption key will be long enough to prevent the exhaustive search. This did not mean the impossibility to find out witch the key was but, as I said before, the impossibility to find out the key in useful time.

One of the main ways of protecting the messages consists in protecting the encryption-decryption algorithm for being published. Thus, the enemy will be in the situation of trying to find both the decryption algorithm and key.

We know that for a successful decryption operation it is based on a well-known decryption algorithm and a decryption key.

One of the main problems that can appear in data reception it is message authentication, made by the receptor. This way, the asymmetrical encryption schemas, using public and private keys, there are algorithms that ensure, by default, this authentication. We can authenticate a message using cryptographic hash functions. Their result will be attached to the main message ([1]). Of course there are a lot of algorithms that can realise these using software techniques.

Basically, the above technique it is used for protecting the message for being affected by errors during the data transmission ([1]). The error detection rate is high (greater than 99.99%) but the transmission could not be absolutely safe.

In this paper we consider an encryption schema that uses a set of encryption transformation $\{E_e: e \in K\}$ and decryption transformations $\{D_d: d \in K\}$, where K is the key space.

Symmetric Encryption

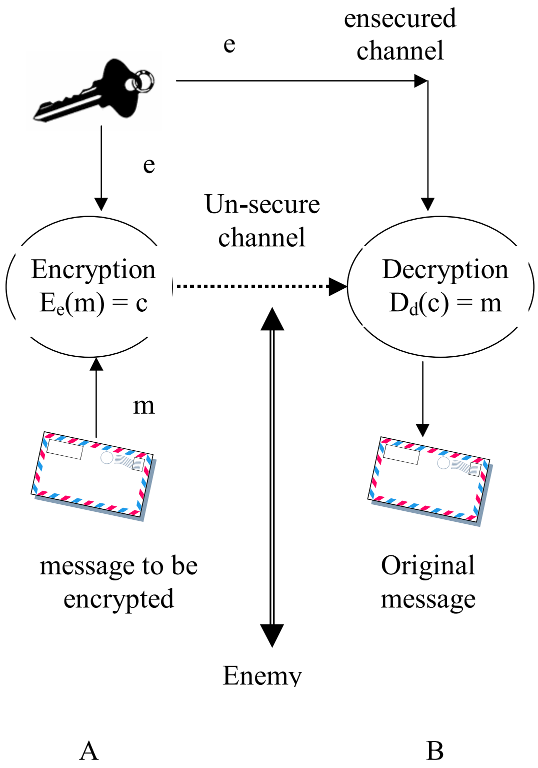

Figure 1. Main communication between the two entities using symmetric encryption and an unsecured channel for key distribution

Encryption it is known as symmetrical if, for every key pair (e,d) it is easy, by the point of view of developing algorithms, to find one key from the other one (in practice, e=d).

There are two main categories for symmetric key encryption, block ciphers and stream ciphers.

A block cipher is an encryption schema that divides the message in blocks with fix length and encrypts each block.

It is one of the most used de symmetric key encryption. It also has two main classes:

- substitution encryption in witch each symbol it is replaced with other symbol or group of symbols;
- transposition encryption, through witch the symbols belonging to a block are permuted.

The key distribution it is made, as observefrom Figure 1, through secured channels.

For simple ciphers, Vigenere type (it uses a cipher block, t length, on an alphabet A, with the property that each element from t defines a permutation), the key space length (the number of combinations that can be made in order to obtain the decryption key) it is $(26!)^3 \approx 10^{79}$.

Public Key Encryption

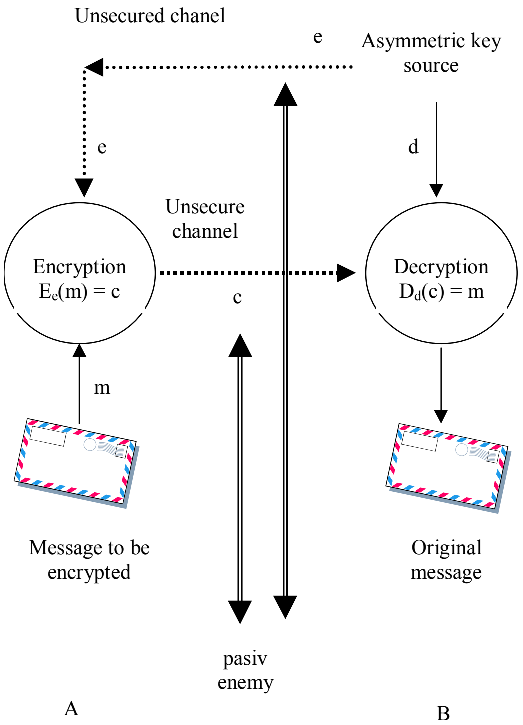

Figure 2. Main communication between the two entities using asymmetric encryption

The asymmetrical encryption principals ([1]), in witch $e \neq d$, consists in generating, by B, for a pair of public and private keys. The public key will be distributed thorough unsecured (but secured preferred !) channels to entities that will encrypt messages designated to B. The received messages will be decrypted only with the B's private key.

According the encryption-decryption schema, described in Figure 2, it appears that the encryption with public and private keys it is safe enough.

If the probability of breaking such an asymmetrical key are smaller than in the case of symmetrical encryption, the schema brings the problem of authentication A in front of B so that B will have the prove that the real sender of the message is A. Through the encryption schema, B will know, using various algorithms, that the message where encrypted with one of his own public keys. Those keys were distributed by B. B will know that the real sender of the message is A only if he gives A a public key through a secured channel. That key were generated especially for A.

It can be reached to the distribution through secured channels of a set of public keys, each one for every emitting entity. That way, the emitter of each message that B receives it is authenticated, through particular algorithms.

Asymmetrical encryption schemas, such like RSA ({1]), Rabin ({1]), etc, implies a series of arithmetic algorithms that works with modulo type operations, power rising (large numbers at a large power).

Symmetric Key Encryption – Asymmetrical Key Encryption Comparison

Basically, symmetric encryption has the advantage of:

- high computing speeds;
- relatively short keys;
- the possibility of grouping them in complicated encryption schemas.

The disadvantages are: the need of frequently changes the keys and their distribution through secure channels.

In the case of public key cryptography, the advantages are:

- only the private key, used for decryption must be made secret;
- the public keys can be distributed through unsecured channels.

Depending on the way of using them, the public and private keys can remain unchanged for long periods of time. Several public key encryption schemas provide digital signature mechanisms. The key used for describing the authentication public function is, normally, smaller than a symmetrical key. In a large network, the number of keys can be smaller than the symmetric key encryption schema.

The disadvantages of public key cryptography are:

- long computing times comparing to the symmetric encryption schemas. This is due to the large number operations that need special arithmetical algorithms.
- the length of the keys is greater than the symmetric key encryption schemas.

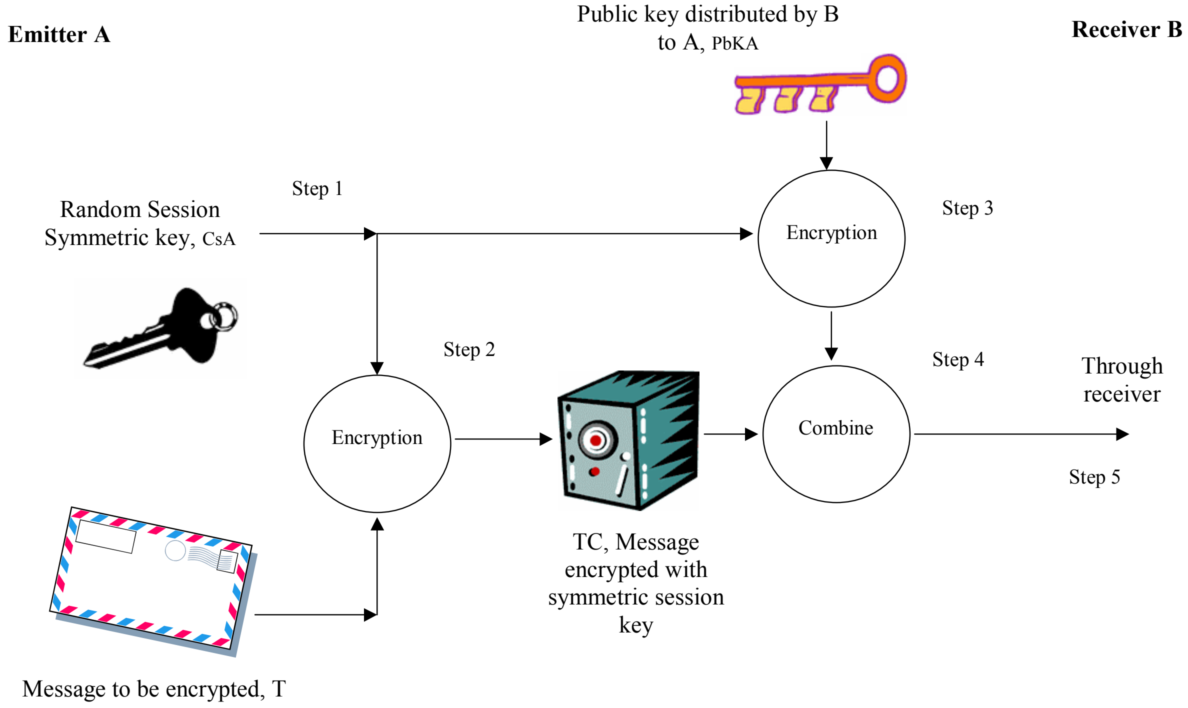
Emitter A
Public key distributed by B to A, PbKA
Receiver B
Step 1
Random Session Symmetric key, CsA
Encryption
Step 3
Step 2
Encryption
Step 4
Combine
Through receiver
Step 5
TC, Message encrypted with symmetric session key
Message to be encrypted, T

Figure 3. Encryption technique. Encryption with symmetric key. Encryption of the symmetric key with a public key.

A System for Encryption Optimisation

A very used technique used for encryption optimisation is based on the principal that smaller text are encrypted with asymmetrical key schemas but long texts are encrypted with symmetric key schemas.

We consider the following practical problem: an entity A must send an encrypted message to an entity B. The requirements are: the encryption operation must be fast, safe and to ensure a *minimum* of authentication.

For encrypting the message we can proceed as show in Figure 3.

Step 1. A session random key, CsA, is generated. This key will be used in symmetric encryption of the text T. The generated symmetric key can have any number of bytes;

Step2. The original message T is encrypted using the session key, CsA;

Step3. Using the public key, received from B, eventually through secure channels, the session key CsA is encrypted;

The order of the steps 2 and 3 can be changed or eventually, those steps can be launched as two separate threads.

Step 4. The resulted streams during the operation from step 2 and 3 where combined and the result were send to destination.

In order to decrypt the original message T, B must follow the next steps, described in Figure 4:

Step 1. Using a splitting algorithm, the streams that contain the ciphered text and symmetric key are separated;

Step 2. Using the private key (PrKB), B can decrypt the symmetric session key (CsA). At this point there are specific algorithm that can ensure the emitter authentication.

Step 3. Using the session symmetric key, described at step 2, the original text can be decrypted.

The problems that appear here are linked to the emitter authentication and original message authentication.

The technique ensures the authentication only of the session symmetric key, CsA. That's because only that one was encrypted with a public key and not the main message.

Of course, an alternative is to compute a check number, using an algorithm based on a hash or polynomial function, at least for the main encrypted message, TC.

Partial authentication of the message

The proposed technique does not consists in authentication of the original message but in a simple algorithm that combines the symmetric encrypted message and symmetric session key

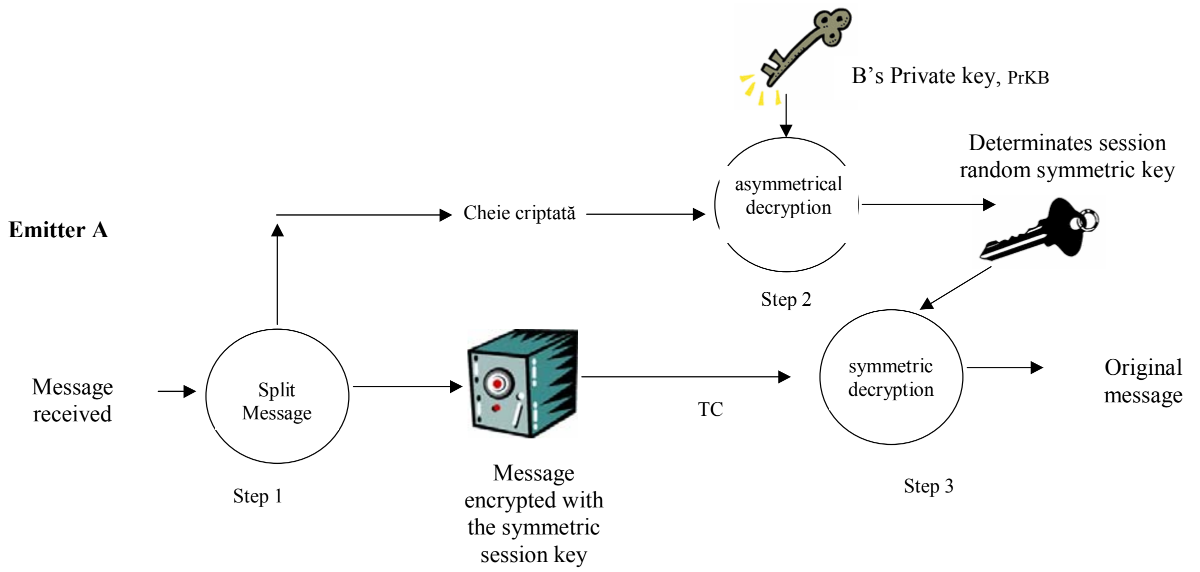

Figure 4. Decryption Technique.

encrypted with a public key.

The proposed algorithm focuses on combining the two parts of the encrypted message to be send to the receiver: the main message (TC in Figure 3 and 4), encrypted with the random symmetric session key and the random session key, encrypted with the public key (referred as CK).

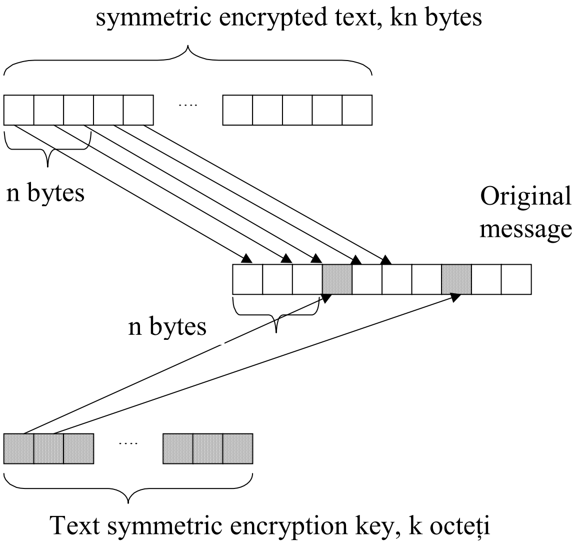
symmetric encrypted text, kn bytes
....
n bytes
Original message
n bytes
....
Text symmetric encryption key, k octeți

Figure 5. The combination of the symmetric encrypted text with the encrypted key

The message authentication is ensured only in cases in witch bytes of the CK are affected. Any other situation is undetectable.

Basically, the contents of the CK will be inserted in the stream TC.

The purpose of the technique is to ensure a protection regarding the separation of the streams CK and TC but also to protect TC against a series of modifications, accidental or not, as a result of an enemy attack.

Lets consider an example in witch the stream TC has kn bytes and CK, k bytes.

As can be observed in Figure 5, in the resulted bytes stream will obtain, at n bytes from TC, one byte from CK.

The purpose was that, for a series of modification for the resulted stream, those would affect the content of minimum one byte of CK so the message receiver will be warned that the message was not authentic.

The liberty degrees that we have in conducting the experiment to test the proposed algorithm are:

- symmetric encryption algorithm,
- length of the session random symmetric key,
- public key encryption algorithm,
- length of the encryption public key,
- length of one block to be symmetrically encrypted (n),
- the algorithm of combine the two resulted streams TC and CK.

The choice of one practical method of combining streams is not critical, in the way that inserting an byte from CK can be made before, after or in any position in the block of the TC's n bytes.

Basic requirements are that, for each block of n+1 bytes from the resulted message, one byte to belong to CK.

Detectable situations in witch the contents of the stream TC is modified are:

- o modification in long series that will continuous affect more than n+1 bytes in the resulted stream. In this case, more than 1 bytes, belonging to CK will be affected and the receiver will detect that;

- only one byte from the resulted stream will be modified. The modification will be detectable only if a byte belonging to the stream CK was modified. The probability that the modification will be detectable will be $\frac{k}{nk+k} = \frac{1}{n+1}$. It can be observed, in this case, that the length of the stream TC negatively affects the probability of detecting the modified byte;

- if several bytes from the resulted stream, in a random order, were modified, the probability of detecting the situation increases with the length k of the session random symmetric key and decreases with the length of the stream TC.

If the size of the stream TC is relatively constant and non-modifiable, we have the possibility of modifying only the size of the stream CK witch is the key for the symmetric key encryption.

Such a large size encryption key for symmetric encryption algorithm provides long encryption times.

The solution is to choose an algorithm with symmetric key encryption in witch the safety will be based, mainly, in the size of the key rather than the algorithm and the number of operations processes by this.

The choice of the encryption algorithm with public key must be made so that provides a safe schema of authentication.

Conclusions

A very common used encryption schema is the one in witch the message is encrypted with a symmetrical key random generated. The resulted stream is referred as TC. That key, at his turn will be encrypted with a public key (the resulted stream will be referred as CK). The two streams will be combined and send to destination.

At the destination, the user will re-create the algorithm so that, using the private key it decrypts the random session key and, with it, decrypts the original message.

Authentication is possible only for that part of the message that contains the key CK.

The encryption schema with partial message authentication consists in inserting bytes of the stream CK in the bytes of the stream TC.

Authentication is ensured in the situations in witch the bytes belonging to CK are affected

The probability of message authentication is directly proportional with the size of session random key and inversely proportional with the size of the encrypted message.

The combining algorithm for the two streams is not critical, while in every n+1 continuous block bytes of the resulted message one belongs to CK.

The symmetric encryption algorithms must be choosing so that ensures the security through the length of the key rather than from a large number of operations.

Public key encryption algorithm must ensure an authentication message schema.

References

[1] Alfred J. Menzenes (1996) *HandBook of Applied Criptography*, CRC Press ISBN: 0-8493-8523-7.

[2] G.B. Agnew, R.C. Mullin, I.M. Onyszchuk, and S.A. Vanstone, "*An implementation for a fast public-key cryptosystem*", Journal of Cryptology, 3 (1991), 63–79.

[3] R. Anderson, "*Searching for the optimum correlation attack*", B. Preneel, editor, Fast Software Encryption, Second International Workshop (LNCS 1008), 137–143, Springer-Verlag, 1995.

[4] M.J. Wiener, "*Efficient DES key search*", Technical Report TR-244, School of Computer Science, Carleton University, Ottawa, 1994. Presentedat Crypto '93 rump session.

[5] M.N. Wegman and J.L. Carter, "*New hash functions and their use in authentication and set equality*", Journal of Computer and System Sciences, 22 (1981), 265–279.