© 2017 IJSRSET | Volume 3 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Maximizing the Security Factor by using Probabilistic Encryption

Masood Ahmad[1] , Ravi Pratap Singh[2]

Department of Computer Science and Engineering, Azad Institute of Management and Technology Lucknow, Uttar Pradesh, India

ABSTRACT

In this paper survey is done corresponding to various algorithm present for converting normal text to encoded form for security reason. Therefore Probabilistic Encryption is selected within this paper to provide secure environment while transmission of talk from one person to another, So as to prevent from third party attacks. Encryption is the process of converting a plain text message in to cipher text which can be decoded back in to the original message. An encryption algorithm along with a key is used in the encryption and decryption of data. Encryption schemes are based on block or stream cipher. In Conventional symmetric encryption a single key is used. With this key the sender can encrypt a message and a recipient can decrypt the message but the security of the key becomes problematic. In asymmetric encryption, the encryption key and decryption key are different. There are several types of data encryption which form the basis of network security. ONE is a public key by which the sender can encrypt the message and the other is a private key by which the recipient can decrypt the message.

Keywords: Probabilistic Encryption, Private Key Vs Public Key, Probabilistic Public Key Encryption, Key Distribution Mechanism

I. INTRODUCTION

Historically, encryption schemes were the first central area of interest in cryptography[18]. They deal with providing means to enable private communication over an insecure channel. A sender wishes to transmit information to a receiver over an insecure channel that is a channel which may be tapped by an adversary. Thus, the information to be communicated, which we call the plaintext, must be transformed (encrypted)to a cipher text, a form not legible by anybody other than the intended receiver.

The latter must be given some way to decrypt the cipher text, i.e. retrieve the original message, while this must not be possible for an adversary. This is where keys come into play; the receiver is considered to have a key at his disposal, enabling him to recover the actual message, a fact that distinguishes him from any adversary. An encryption scheme consists of three algorithms: The encryption algorithm transforms plaintexts into cipher texts while the decryption algorithm converts cipher texts back into plaintexts.

A third algorithm, called the key generator, creates pairs of keys: an encryption key, input to the encryption algorithm, and a related decryption key needed to decrypt. This work mainly deals with the algorithm which generates sub keys which provides sufficient strength to the encryption mechanism. Any symmetric encryption scheme uses a private key for secure data transfer. In their work on " A simple algorithm for random number generation [7], Encryption of information is the most common means of providing security.

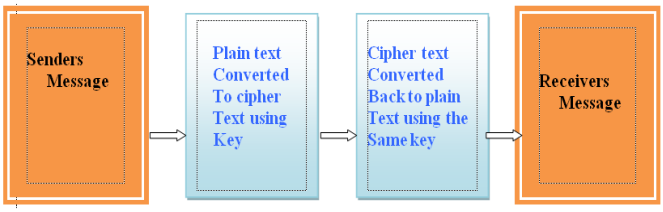

Figure 1 : Encryption Model

This general model shows that there are four basic tasks in designing a particular security service.

- Designing an algorithm for performing encryption & decryption process.

- Generating the secret information with the help of algorithm of step 1.
- Identifying methods for the distribution and sharing of secret information.
- Identifying rules to be used by both the participating parties to make it secured.

A crypto system is an algorithm, plus all possible plain texts, cipher texts and keys. There are two general types of key based algorithms: symmetric and public key. With most symmetric algorithms, the same key is used for both encryption and decryption, as shown in Figure 2.

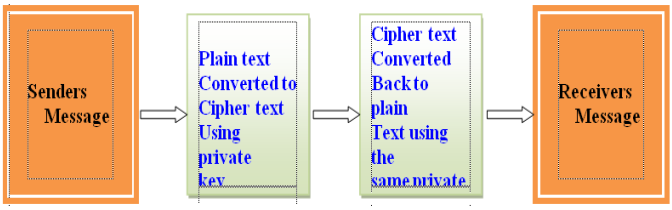

Figure 2: Symmetric-Key Encryption

The process of symmetric-key encryption can be very fast as the users do not experience any significant time delay because of the encryption and decryption. Symmetric-key encryption provides security to data as the key is shared only by the participating parties. It also provides a degree of authentication, since information encrypted with one symmetric key cannot be decrypted with any other symmetric key. Thus, as long as the symmetric key is kept secret by the two parties using it to encrypt communications, each party can be confident that it is communicating with the other as long as the decrypted messages specify a meaningful sense.

II. METHODS AND MATERIAL

Definition 1 [Probabilistic Algorithm]. A probabilistic algorithm is an algorithm with an additional command RANDOM that returns "0" or "1", each with probability 1/2. In the literature, these random choices are often referred to as **coin flips**.

A. Private-Key vs. Public-Key

The first encryption schemes had only one key for encryption and decryption, which therefore was to be kept private. We call them **private-key** or symmetric encryptions chemes. Before using the scheme, the key must once be exchanged securely, hence private-key encryption is a "way of extending a private channel over time".

In the 1970s Diffie and Hellman [1] introduced a new concept, called publickey or asymmetric encryption: The encryption key differs from the decryption key, moreover, given the former it must be infeasible to find the latter. That is why these schemes provide secure communication without ever requiring any private channel; the receiver creates a pair of keys, gives the encryption key (that can be publicly known) to the sender, but keeps the decryption key secret. The sender can then use the encryption key to encrypt messages, which can only be decrypted by the receiver. In this context, the encryption and the decryption key are often referred to as the public and the private key, respectively.

An analogy illustrating the difference between the two notions of encryption is the problem of sending a confidential parcel by postal delivery: Private-key encryption corresponds to sending the secret content in a locked box. The drawback is that the recipient must be given the key to the box. A secure channel, e.g. a courier, is thus required. The idea to avoid this is not to send the key to the receiver, but rather let him send a padlock to the sender and keep the key. The sender locks the secret content and the receiver is the only one able to open the box. Intercepting the padlock is of no use to an adversary - assuming that it does not help in forging the key.

B. Probabilistic Public-Key Encryption

The first public-key cryptosystems (such as RSA [4]) were deterministic algorithms based on trapdoor functions. These are functions that are easy to compute but hard to invert—unless some information called the trapdoor is known. So, while everybody can use the function to encrypt messages, only the legal receiver knows the trapdoor, which serves as a decryption key.
According to [2], the two main drawbacks of encryption schemes based on trapdoor functions are:

1. Inverting may be easy for plaintexts of some special form.
2. It could be easy to compute at least partial information of the plaintext.

Furthermore, for deterministic schemes it is easy to detect if a message is sent twice. These points inspired the development of probabilistic public-key encryption schemes by Goldwasser and Micali [2]. They substituted the notion of trapdoor functions by what they introduced as (unapproximable) trapdoor predicates : A predicate B is trapdoor and unapproximable if anyone can select an x such that $B(x) = 0$ or y such that $B(y) = 1$, but only those who know the trapdoor information can, given z, compute the value of B(z). Goldwasser and Micali used the predicate "is quadratic residue modulo composite n" (see Section 4).

Their scheme uses bitwise encryption, which depends on a sequence of random bits. However, messages are always uniquely decryptable. Two properties are:

1. Decoding is easy for the legal receiver of a message, who knows the trapdoor information, but provably hard for an adversary.
2. No information about the plaintext can be obtained from the ciphertext by an adversary.

Definition 2. [Probabilistic Public-Key Bit-Encryption Scheme]. A probabilistic public-key bit-encryption scheme (K, E, D) with security parameter n consists of :

- K, the key Generator : A probabilistic algorithm that on input n outputs a pair (e, d), where e is the public key and d is the private key.
- E, the encryption function, with three inputs: the public key e, the plaintext bit b 2 {0, 1}, and a random string r of length p(n) for some polynomial p(·). We will write Ee(b, r).
- D, the decryption function, with two inputs: the private key d and the ciphertext c. Again, we will write Dd(c).

C. Deterministic and Probabilistic Encryption

- Deterministic is a specific type of encryption. In this type of encryption, the resulting converted information, called ciphertext, can be repeatedly produced, given the same source text and key. For example, if you know that the message 'hello world' has the ciphertext '&yy/ m/jyp' under some form of deterministic encryption, then that message will always produce the same ciphertext.
- **Probabilistic** is also a specific type of encryption. But unlike deterministic, it introduces an element of chance. Source text repeatedly encrypted with the same key will normally yield different ciphertext. So, a simple message like 'hello world' won't always correspond to the same ciphertext. Instead, that message would produce one of many possible ciphertexts each time it's encrypted.

III. RESULTS AND DISCUSSION

Key Distribution Mechanism

In most of the schemes, a key distribution centre (KDC) is employed which handles the task of key distribution for the participating parties. Generally two mechanisms are employed.

In the first mechanism user A, requests KDC for a session with another user say, B. Initially the KDC sends session key encrypted with private key of A, to the user A. This encrypted session key is appended with encrypted session key by private key of B. On receiving this User A, gets session key and encrypted message with private key of B. This encrypted message is sent to B, where B decrypts it and gets the session key. Now both A & B are in hold of session key which they can use for secured transmission of data. Otherwise it is the KDC which sends encrypted session key to the participating parties based on the request of user.

In the second mechanism, the scenario assumes that each user shares a unique master key with the key distribution centre. In such a case, the session key is encrypted with the master key and sent to participating parties.

A more flexible scheme, referred to as the control vector [10]. In this scheme, each session key has an associated control vector consisting of a number of fields that specify the uses and restrictions for that session key. The length of the control vector may vary. As a first step, the control vector is passed through a hash function that produces a value which is equal to encryption key length. The hash value is XOR ed with the master key to produce an output that is used as key to encrypt the session key. When the session key is delivered to the user the control vector is delivered in its plain form. The session key can be recovered only by using both master key that the user shares with the KDC

and the control vector. Thus the linkage between session key & control vector is maintained. Sometimes keys get garbled in transmission. Since a garbled key can mean megabytes of unacceptable cipher text, this sis a problem. All keys should be transmitted with some kind of error detection and correction bits. This is one way errors of key can be easily detected and if required the key can be reset.

IV. CONCLUSION

In this paper survey is done on a method of dense probabilistic encryption which has many similarities to, but many advantages over, the original method of probabilistic encryption introduced by Goldwasser and Micali. These advantages also apply relative to all previous methods of probabilistic encryption. Many Variations of this method are possible depending on the user's willingness to depend upon stronger assumptions in exchange for more efficient decryption. Applications of this method have been also given in which traditional encryption methods are not just less efficient, but are instead unusable.

V. REFERENCES

[1]. W. Diffie, M. Hellman, New Directions in Cryptography, IEEE Transactions on Informations Theory, IT-22(6), pp. 644–654, 1976

[2]. S. Goldwasser, S. Micali, Probabilistic Encryption, Journal of Computer and System Sciences, 28, pp. 270–299, 1984

[3]. A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, 1996

[4]. R. Rivest, A. Shamir, L. Adleman, A Method for Obtaining Digital Signature and Public Key Cryptosystems, Communications of the ACM, 21(2),pp. 120–126, 1978

[5]. Guo D, Cheng L.M., Cheng L.L: A New Symmetric Probabilistic Encryption Scheme Based on Chaotic Attractors of Neural Networks, Applied Intelligence, Vol 10, No.1, Jan 99, pp 71-84.

[6]. G¨otz, M., Kelber, K. & Schwarz, W. 1997"Discrete-time chaotic encryption systems–Part I: Statisti-cal design approach,"IEEE Trans. Circuits andSystems–I 44(10), 963–970.

[7]. Carlone Fontaine & Fabien Galand: A Survey of Homomorphic Encryption for non Specialists, EURASIP Journal, Vol 07, Article 10.

[8]. Donavan G.Govan, Nathen Lewis: Using Trust for Key Distribution & Route Selection in Wireless Sensor Networks, International Conference on Network Operations & Management, IEEE Symposium 2008, PP 787-790.

[9]. Dorothy E. Denning et al.: Time Stamps in Key Distribution Protocol, Communication of ACM, Vol 24, Issue 8, Aug 1981, pp 533-536.

[10]. E. C. Park, I.F.Blake: Reducing communication overhead of Key Distribution Schemes for Wireless SensorNetworks: Computer Communications & Networks, ICCCN 2007, pp 1345-1350.

[11]. Georg J. Fuchsbauer: An Introduction to Probabilistic Encryption, "Osjecki Matematicki List 6(2006), pp37-44.

Author's Profile

Ravi Pratap Singh pursuing M.tech from Computer Science in 2016 from Azad institute of Management And Technology Lucknow. Completed B.tech in Information Technology in 2013 from Accurate Institute of Management And Technology Gr .Noida Currently Working as an Android Developer in NIIT Technologies . My area of interest cryptography.

Masood Ahmad working as an Assistant Professor (Computer Science Dept) Azad institute of Management and Technology Lucknow. Completed B.tech in Information Technology in 2008 from Babu Banarasi Das National Institute of Technology & Management Lucknow .Completed M.tech in computer Science in 2014 from Azad institute of Management And Technology Luck now.