International Journal of Soft Computing 1(3): 220-224, 2006

Review on Session-Keys and Their Importance for Secured Electronic Transactions

M. Ismail Jabiullah and M. Lutfar Rahman
Department of Computer Science and Engineering, Dhaka University, Bangladesh

Abstract: Security of electronic transactions in today's high-speed electronic connectivity is a very important matter. The explosive growth in computer networks has increased the dependence of organizations and individuals on the information through network transactions. The trend to make security of the electronic transactions is now a principal interest. Cryptography is the heart of the secured electronic transactions. It is becoming increasingly important as a fundamental building block of network security by using the secret session-key. Session-keys thus play the central role to enhance network security and so it is the most important aspect of today's high-speed electronic communication connectivity. In this study, session-keys and their importance, lifetime and the application areas are focused briefly in the secured electronic communication session's aspects.

Key words: Session-keys, secured electronic, high-speed

INTRODUCTION

In today's global electronic connectivity through the Internet, security of electronic transactions is an important and fundamental matter[1]. Several approaches are engaged to ensure the security for the increasing demand of the electronic transactions. Cryptography plays the fundamental role in the security of network communications. Cryptographic mechanisms are based on the stronger secret-key and the security related transformations. All the mechanisms are designed in such a way that they can protect all sorts of network attacks, e.g., eavesdropping, masquerade, replay attacks, denial of services and modification of message. Secret-key provides a secured communication among all the communicating parties involved a transaction or a session. Each transaction performs a communication to deliver a message or information with a strong transformation using stronger secret-key[2]. A session-key is a secret key that is just used for one communication session and after the communication, it is discarded. There is no reason to store the session-keys after they have been used. Session-keys are useful because they only exist for the duration of a particular communication session. In many cases, frequent key changes are desirable to enhance the better security in the transactions using the session-keys. Motivations for the use of secret session-keys include the following:

- To limit available ciphertext under a fixed key for cryptanalytic attack.
- To limit exposure, with respect to both time period and quantity of data, in the event of session-keys compromise.
- To avoid long term storage of a large number of distinct secret session-keys in the case where one terminal communicates with a large number of others, by creating keys only when actually required.
- To create independence across communication sessions or applications.

Electronic transactions are fairly obvious that if the used secret key length were not sufficiently large, the cryptographic communication schemes would not be secured because it would be easy to search through all possible secret keys. For that reason, cryptographic secret keys are taken as long as the used block length of the communicating plaintext message. But these are not secured all alone[3]. If a secret session-key is used for long time communication, someone may hijack the entire session-key impersonating the network address of one of the communicating parties after the initial authentication is completed. So, frequent key changes are necessary to establish a secured transaction. In this study, secret session-key, its importance, functions, applications, distribution, lifetime, attacks and countermeasures are the main discussing areas.

Types of cryptographic keys: A cryptographic algorithm is the mathematical function used for encryption and decryption. Cryptography performs the encryption with the key value that might be any one of a large number[3].

Corresponding Author: M. Ismail Jabiullah, Department of Computer Science and Engineering, Dhaka University, Bangladesh

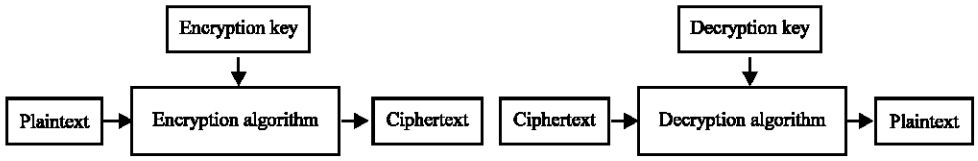

Fig. 1: Symmetric-key

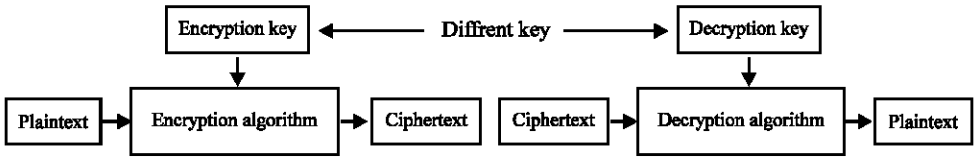

Fig. 2: Asymmetric-key encryption/decryption

The range of possible values of the key is the key space. Both the encryption and decryption operations use this key and function became $E_K(P) = C$ and $E_K(P) = C$, where P is the plaintext, K is the key, E is the encryption algorithm, C is the cipher text and D is the decryption algorithm. There are two types of keys used encryption and decryption algorithms. If the same key is used in encryption and decryption algorithms, the key is called symmetric-key or the private-key algorithm (Fig. 1). In symmetric-key cases, it is required that both the sender and the receiver agree on a key before they can communicate securely.

The sender uses the key and an encryption algorithm to encrypt the data message and receiver uses the same key and the corresponding decryption algorithm to decrypt the received encrypted data message. The decryption algorithm is the inverse of the algorithm used for encryption. Symmetric-key algorithms are efficient; it takes less time to encrypt the plaintext message. The reason is that the key is usually smaller and it takes less mathematical computation. And so the symmetric-key is used to encrypt and decrypt long messages[4].

If the encryption algorithm and the decryption algorithm uses the different key, the keys are called public-private key or the public key or the asymmetric-key algorithm (Fig. 2).

In the public-key cryptography, two keys, one is private and the another is public, are used both for encryption in the sender end and for decryption in the receiver end. The sender encrypts the message with his/her private key and the receiver's public-key to produce the ciphertext. On the other hand, the receiver decrypts the received ciphertext with his/her private key to retrieve the plaintext from the received ciphertext. In public-key encryption/decryption process, each party creates a pair of keys, the private one kept and the public one is distributed publicly. Each party is independent and the pair of keys created that can be used to communicate with any other party. For an effective method, the public key algorithm needs large numbers and so calculating the ciphertext from the plaintext using long keys takes a lot of computing time[5]. For that reason, public-key algorithms are more efficient and secured for short message communications especially for symmetric-key or secret session-key distributions.

Table 1: Session-key Size, key space and required time

Key size (bits)	No. of alternative keys	Time required for a 1encryption/μs	Time required at 10^6 operations/μs
32	$2^{32} = 4.3\times10^{9}$	2^{31} μs = 35.8 Minutes	2.15 milliseconds
56	$2^{56} = 7.2\times10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4\times10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7\times10^{50}$	2^{167} μs = 5.9×10^{36} years	6.4×10^{30} years
256	$2^{256} = 1.2\times10^{77}$	2^{255} μs = 1.8×10^{63} years	1.8×10^{57} years
512	$2^{512} = 1.3\times10^{154}$	2^{511} μs = 2.1×10^{140} years	2.1×10^{136} years
1024	$2^{1024} = 1.8\times10^{308}$	2^{1023} μs = 9.0×10^{307} years	9.0×10^{301} years
2048	$2^{2048} = 3.2\times10^{616}$	2^{2047} μs = 5.1×10^{602} years	5.1×10^{598} years

Secret-key cryptography involves the use of a single key. It is fairly obvious that if the key length were not sufficient large, the cryptographic schemes would not be secured because it would be too easy to search through all possible keys. For that reason, secret keys are taken as along as the used block length of the plaintext message. Several key size, key space, time required for an encryption per microsecond and the time required a 10^6 operations per microsecond are presented below. Assuming that for each key size takes 1 microsecond to perform a decryption operation. The results for system that can process 1 million keys per microsecond are considered and are shown in Table 1.

But this is not secured all alone. If a secret key is used for a long time communication some one may hijack the entire session by impersonating the network address of one of the communicating parties after the initial authentication is complete. To solve this, a key is needed

that is restricted to a short time period such as a single communication or a session, which is, called a session-key. After each communication session, all trace of it is eliminated[6].

The more frequently session-keys are exchanged, the more secured they are, because the opponent has less ciphertext to work with for any given session-key. But the distribution of session-keys delays the start of any key exchange and creates a huge burden of the network capacity of the communication. A key distribution manager must try to balance these competing considerations to set the lifetime of a particular session-key. In connection-oriented protocols the same session-key is used for the length of time till the connection is open. A new session-key is used in each new session. If a logical connection has a very long lifetime, then it will be prudent to change the session-key every time[7]. In connectionless protocols, where transaction-oriented process is happened, there is no explicit connection initiation or termination in the route. So, one need not to be changes the session-key very often. The most secured approach is to use a new session-key for each exchange[8]. Here a better strategy is that a given session-key is used for a certain fixed period only or for a certain number of transactions.

Importance of session-key: Session-keys provide many information security functions such as authentication source and message, integrity and privacy. The secret session-keys are employed to establish secured electronic transactions in the following areas.

Digital signature: Public key cryptographic techniques require a signer to generate a unique secret number as a session-key for each communicating message. If the same session-keys were used for two different messages, it would expose the signer's session-key[9]. If a secret session-key were predictable or guessable, the signer's secret session-key would be exposed. So, the session-key is used in per message in electronic communication to overcome the situation and to unforge the signatures.

Strong authentication: If all that was done about network security were to replace all the plaintext exchanges with cryptographic authentication, computer networks would be a lot more secured than they in today. But there are security vulnerabilities that occur after authentication. An eavesdropper might steal information by seeing the conversation. Something along the path, e.g., a router, might intercept messages in transit and modify them or replay them, or someone on the path might hijack the entire session by impersonating the network address of one of the parties after the initial authentication is complete[10]. One can protect against eavesdroppers, session hijackers and message manglers by using cryptographic secured session-key throughput the conversation or communication.

Kerberos authentication: In Kerberos authenticator, each ticket includes a session-key that is used by the client to encrypt the authenticator sent to the service associated with the ticket. The client and the server to protect messages passed during that session may subsequently use the session-key.

Security handshake: Cryptographic mutual authentication is certainly an improvement, but it is also important to cryptographically protect the transactions after the initial handshake. It is obviously desirable to protect the data from disclosure or modification. But another vulnerability if cryptographic protection ends after the initial handshake is session hijacking, in which someone takes over sender's session to receiver by forging sender's IP address as the secure address on packets sent to receiver and using TCP sequence number larger than what sender would be using. Without cryptographic protection, receiver can not distinguish these packets from authentic packets from the sender. Once receiver accepts the attacker's larger TCP sequence numbers, sender's data just gets ignored as duplicate data. It looks to sender like the connection breaks, but the attacker is now logged in as sender and can do anything sender is allowed to do. So, a session-key is used after the initial mutual authentication to cryptographically protect the conversation from disclosure, modification or hijacking.

Pretty Good Privacy (PGP): In PGP, each session-key is associated with a single message and is used only for the purpose of encrypting and decrypting that message. It is mentioned that message encryption /decryption is done with a symmetric key encryption algorithm. Random 128-bit numbers are generated by CAST-128 itself. The input to the random number generator consists of a 128-bit key and two 64-bit blocks that are treated as plaintext to be encrypted. Using CFB mode, the CAST-128 encrypter produces two 64-bit ciphertext blocks, which are concatenated to form the 128-bit session-key.

Session-key distribution process: In a trusted key distribution center, each party in the network shares a secret key, known as master key, with the KDC. The KDC is responsible for generating session-keys to be used for a short time over a connection between two parties and distributing those session-keys using the master keys to protect the distribution[11].

Control vector encryption and decryption for session-key distributions: The main drawback of the technique is that one can easily analyze the key, K without knowing both the control vector CV and the master key K_m by hitting the encryption function. This drawback is overcome by the modified technique, where the key is hiding from the intruders by increasing the level of encryption. Thus it became more secure and reliable. The session-key K_s is to be distributed through the KDC using the control vector encryption/decryption process. The control vector CV is considered as a pseudorandom bit string whose length is not fixed. The master key K_m is considered as a fixed length pseudorandom bit string whose length is 128. The control vector is passed through hash function MD5 that produces a hash value h whose length is also 128, equal to the length of the master key K_m. The control vector CV is inputted to the hash function MD5. The hash output and the session-key K_s is used in the encryption technique Data Encryption Standard (DES). The encrypted output is again encrypted using the same session-key K_s. Then the XORed output of the session-key K_s and the hash output and the previously encrypted output are again inputted to the encryption function DES[12]. The resultant of the encryption function is then transmitted to the destination. In the receiving end, the control vector is inputted to the hash function MD5 and the hash output and the K_m is XORed and inputted to the decryption function DES with the received encrypted session-key EKs. The decrypted output is again decrypted using the DES with the master key K_m. Finally, the distributed session-key K_s is found by decrypting the previous calculating result with the hash output[13].

Cryptanalysis of session-keys: The whole point of cryptography is to keep the plaintext or the key or the both as the secret fashion from eavesdroppers. Eavesdroppers are assumed to have complete access to the communications between the sender and the receiver. Successful cryptanalysis may recover the secret session-keys of the plaintext. An attempted cryptanalysis is called an attack. The secrecy must reside entirely in the communicating keys. There are seven general types of cryptanalytic attacks. They are ciphertext-only attack, known-plaintext attack, chosen-plaintext attack, chosen-ciphertext attack, adaptive-chosen-plaintext attack, chosen-key attack and purchase key attacks. Most cryptosystems are breakable in ciphertext-only attack, simply by trying every possible key one by one and checking whether the resulting plaintext is meaningful. This type of attack is called brute force attack. Cryptography is more concerned with cryptosystems that are computationally infeasible to break and this type of system is considered as computationally secured, since it cannot be broken with available resources, either current of future.

One can measures the complexity of an attack in the following types:

- Data complexity where the amount of data needed as input to the attack
- Processing complexity where the time needed to perform the attack and
- Storage requirement where the amount of memory needed to do the attack.

Complexities are expressed as orders of magnitudes. If an algorithm has a processing of complexities of 2^{128}, then 2^{128} operations are required to break the algorithm. While the complexity of an attack is constant, computing power is anything but. There have been phenomenal advances in computing power during the last half-century and there is no reason to think this trend would not continue. Many cryptanalytic attacks are perfect for parallel machines. Good cryptosystems are designed to be infeasible to break with the computing power that is expected to evolve many years in the future.

CONCLUSION

Secret session-key encryption/decryption process is an important matter for secure electronic transactions. By frequent key changes a common cryptographic technique is to encrypt each individual conversation with a separate session-key. Session-keys are useful and thus ensure better security because they only exist for the duration of one telecommunication conversion. Session-keys are implemented in most of the application areas of the network security. Many of them are presented and discussed. So, it is realized that the more frequently session-keys are exchanged, the more secured they are.

REFERENCES

1. William, S., 2003. Cryptography and Network Security. Principles and Practice, 3rd Edn., Pearson and Education.
2. Charlie, K., R. Perlman and M. Speciner, 2003. Network Security. Private Communication in a Public World, 2nd Edn., Pearson and Education.
3. Menezes, A., P. Van Oorschot and S. Vanstone, 1997. Handbook of Applied Cryptography. CRC Press.
4. Bruce, S., 1996. Applied Cryptography. 2nd Edn., John Wiley.
5. Lutfar M.R., 2004. A review on Cryptography and Cryptographic Applications. Magazine, Department of Computer Science and Engineering, 1st Edn., Dhaka University.

6. William, S., 2003. Data Communications and Computer Network. 6th Edn.
7. Andrew, T., 2003. Computer Network. 4th Edn.
8. Behrouz, A.F., 2004. Data Communications and Networking, 3rd Edn., Tata-McGraw Hill.
9. Ismail, M.J., Sk. Mizanur Rahman and M. Lutfar Rahman, 2002. Session-key Generation for Message Authentication using Conventional Encryption Techniques. Proceedings of the 3rd International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT'02, Kanazawa, Japan.
10. Ismail, M.J., Sk. Mizanur Rahman and M. Lutfar Rahman, 2004. Pseudorandom bit string for Cryptographic Applications. J. Sci., Dhaka University, ISSN 1022-2502.
11. Ismail, M.J., Sk. Mizanur Rahman and M. Lutfar Rahman, "Encryption with Randomly Chosen Base Conversion and Special Symbols", Accepted for publication, "Nuclear Science and Applications", J. Bangladesh Atomic Energy Commission, Dhaka.
12. Ismail, M.J., Abdullah Al-Shamim and M. Lutfar Rahman, 2003. Session-key Generation using Conventional Encryption Techniques. Proceedings of the 6th International Conference on Computer, Communication and Information Technology, ICCIT'03, Jahangir Nagar University, Bangladesh.
13. Ismail, M.J., Sk. Mizanur Rahman and M. Lutfar Rahman, 2001. Session-key Generation for Message Authentication using Conventional Encryption Techniques, Journal of Electrical Engineering, The Institute of Engineers, Bangladesh.