Chapter

Secure Communication Using Cryptography and Covert Channel

Tamer S.A. Fatayer

Abstract

The keys which are generated by cryptography algorithms have still been compromised by attackers. So, they extra efforts to enhance security, time consumption and communication overheads. Encryption can achieve confidentiality but cannot achieve integrity. Authentication is needed beside encryption technique to achieve integrity. The client can send data indirectly to the server through a covert channel. The covert channel needs pre-shared information between parties before using the channel. The main challenges of covert channel are security of pre-agreement information and detectability. In this chapter, merging between encryption, authentication, and covert channel leads to a new covert channel satisfying integrity and confidentiality of sending data. This channel is used for secure communication that enables parties to agree on keys that are used for future communication.

Keywords: encryption, authentication, dynamically, covert channel, confidentiality, algorithm, undetectability, fake key

1. Introduction

Encryption is considered the main key factor of security to achieve confidentiality and to protect data from disclosure [1]. Encryption is not efficient to achieve integrity. It needs another factor called authentication [2]. Covert channel is created to transfer data indirectly between client and server; it was created by Lampson [3].

Before the client and server use the covert channel, they must agree on a pre-agreement knowledge. Also, they agree on how to send that knowledge. A good example is they agree on even word meaning "00" and odd word meaning "11." If the client sends "communication channels," the server will know that the client's message is "1100" [4, 5].

Covert channel cannot be detected if the following two factors exist: plausibility and bit distribution. Plausibility deceives the attacker who thinks that the channel is normal channel and it is not used to send secret information. On the other hand, the bit distribution of normal channel must same distribution of covert channel [5].

The technique is worked as shown in **Figure 1**, where it shows the general idea of the proposed technique. Covert channel needs shared information. In my protocol, it is considered as a table shared between the client and server. This table contains characteristics of the client such as the name which represents the original key. Each

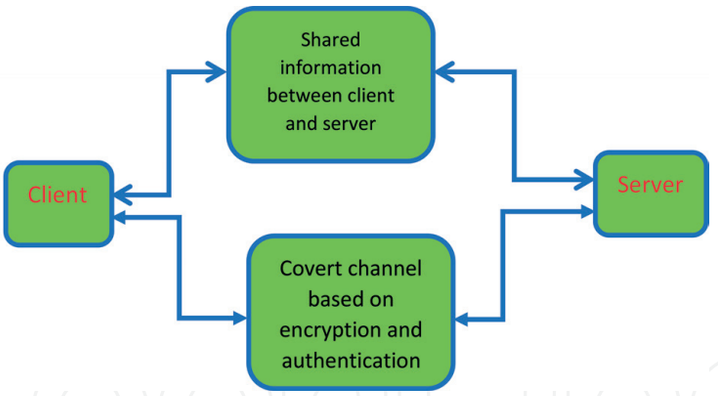

Figure 1.
Secure communication using covert channel, encryption, and authentication.

original key has fake keys. I used encryption algorithm to guarantee the confidentiality. HMAC is used to check integrity. Finally, the time that is needed for the client and server to agree on secret information (e.g., secret keys) is measured.

In this chapter, secure communication channel for transferring data is implemented. The channel between the client and server is considered a covert channel that depends on authentication and encryption.

2. Background

Lampson was the first to introduce the idea of a covert channel [3]. Transferring data between two entities indirectly through a channel is called a covert channel. Before the client and server use the channel to transfer data, they must agree on a pre-agreement knowledge (e.g., shared memory, table). For example, a word containing "mm" means bit "0" other than this means bit "1." So, if the client wants to send "10" to the server indirectly, the client will send "secure communication" to the server. The attacker hardly breaks the covert channel and it is considered to be more secure if it is undetectable [3, 4].

2.1 Covert channel characteristics and properties

Although a covert channel transfers information in a hidden way, it has the same characteristics as other communication channels. These characteristics are:

- Capacity: the amount of data that can be transmitted through the channel. From security viewpoint, increasing channel capacity leads to more information leakage. The covert channel capacity is measured in bits/second. To obtain maximum bandwidth through a covert channel, encoding schemes must be chosen between the sender and receiver.

- Noise: transmitted data through a covert channel are exposed to an amount of perturbations that makes the transmitted and received information between two entities not the same.

- Transmission mode: the transmission of information in covert channels (as in normal channels) can be synchronous or asynchronous. The sender and receiver in synchronous mode should manage their transmission based on a condition or a specific event. On the other hand, in asynchronous mode, the transmission occurs without a prior condition.

2.2 The covert channel is more private and undetectable if it satisfies the following

1. Plausibility: the TCP is usually used for Internet traffic, and it always employs using time stamp option. As a result, TCP using time stamp is a plausible covert channel because the majority of users using TCP will not use it for sending covert data. So, the adversary will believe that TCP time stamps will not be used for sending data covertly.

2. Undetectability: in order for a channel to be more undetectable, the channel must satisfy that the distribution of bits with covert data must be similar to the distribution of the normal channel. If an adversary notices that there are differences (using statistical tests) in bit distribution, then he will detect that the channel is a covert channel. Also, to achieve undetectability, the channel's bits must be random; otherwise, it will be noticed by the adversary.

3. Indispensability: Lampson [3] reports that a communication channel is a covert channel if it is neither designed nor intended to transfer information at all. The channel should introduce several benefits to the users besides sending data covertly; thus, the adversary cannot or will not close off that channel.

2.3 Covert channel classification

Covert channels can be classified as storage or timing channels, noisy or noiseless channels, and program-flow channels.

2.3.1 Storage channels and timing channels

The covert storage channel depends on a shared variable or a storage location, whereby one process (sender) can be allowed to write directly or indirectly to the storage location and the other process (receiver) reads from that storage location. On the other hand, the covert timing channel enables senders to send information to the receiver through signals, whereby the sender manages the time that is needed to perform some operation in such a way that when the receiver observes the time, it will understand a special event or a special piece of information. The main disadvantage of the timing channel is that it is considered very noisy because of the several external factors that affect the execution time of a process. Covert storage channels and timing channels need a synchronization process, which enables the sender and receiver to synchronize with each other to send and receive information. The storage covert channel uses a data variable to enable the sender and receiver to communicate. Therefore, a synchronization variable, called sender-receiver, is needed by the sender to notify the receiver that he has completed reading or writing a data variable. The covert channel uses another synchronization variable, called receiver-sender. To distinguish between storage and timing channels, if a channel uses a storage variable to transfer data between the sender and receiver, it is considered a storage channel. On the other hand, a covert timing channel uses time reference (e.g., a clock) to transfer data between the sender and receiver, whereby the sender and receiver use a common time reference.

2.3.2 Noisy and noiseless channels

I discussed previously that the characteristics of the covert channel are similar to any communication channel. One of these characteristics is that the channel may be

noisy. The covert channel can be noiseless if the transmitted data by the sender and received data by the receiver are the same with probability 1; otherwise, the channel is noisy. Usually, data transmitted through a covert channel is represented by bit "0" or "1." Nevertheless, if the receiver decodes every bit transmitted by the sender correctly, then the covert channel is considered noiseless. Thus, to reduce error rate, which is produced by noise, correction codes are used [6].

2.3.3 Program-flow channels

I present a new type of covert channel, which is program-flow. The program-flow covert channel depends on the flow of program execution to convey information. In our proposed covert channel, the sender tries to guess the correct delta_mmap (encoded information) of the vulnerable server program. The server code which executes in case of successful guess differs from which executes in a failed guess. The receiver distinguishes between server code executed in successful and failed guesses.

2.4 Authentication and key exchange

Authentication process identifies entities that are attempting to access some resources. Diffie-Hellman (DH) algorithm is used as method of public key exchange.

2.4.1 Authentication process

Authentication is a process of checking whether someone or something is authorized or not to access some resources. Authentication can be computer to computer or process to process and mutual in both directions [7, 8]. Bob can authenticate Alice's identity depending on four factors [7, 9], which are:

2.4.1.1 Something you know

Alice sends a request to the server to access some resources; Bob authenticates Alice by asking her about a secret thing that she knows, such as password. If Alice issues a correct password, then Bob will accept her request for accessing some resources. Fortunately, a password is needed to login into the system and access its resources. Yet, unfortunately, the user is always asked to reuse the password when he wants to log into the system, which gives attackers opportunities to hack the password and reuse it. The solution for this problem is to use a onetime password (OTP) so that the user each time she logs into a system needs a new password.

2.4.1.2 Something you have

One of the disadvantages of the first authentication factor (something you know) is that the user may forget his password. Thus, the second authentication factor (something you have) overcomes this problem, whereby the user has an object (e.g., automatic teller machine (ATM) cards, OTP cards [7], and smart cards [9]) to access the system. Unfortunately, the objects may get stolen by attackers.

2.4.1.3 Something you are

The third authentication factor is based on the measurements of the user's physical characteristics such as the fingerprints, iris, and voice. The techniques that measure the behavioral characteristics of the user are called biometrics [7, 8]. This

factor overcomes the problems of the previous factors because it does not depend on a password or a token.

2.4.1.4 Somebody you know

Brainard et al. [9] proposes a fourth factor of authentication that is dependent on emergency authenticator, and it is used when the primary authenticator is unavailable to a user. A good example for emergency system is email; thus, when a user forgets his password, he often has the option of having password reset instructions. A system called "vouching" is introduced. A voucher system permits swapping of the roles of the token and PIN to deal with the case when the user has forgotten his PIN but still has his token.

2.4.2 Message authentication

When Alice and Bob want to exchange messages, they do not want an attacker to modify the contents of their messages. This can be achieved by using message authentication odes (MACs), where the MAC is a tag, attached to the message by Alice to Bob or vice versa. If Bob validates this tag, the request of Alice will be accepted by Bob; otherwise, it is rejected [7]. MAC that is based on cryptographic hash functions is called HMAC [10]. There are many hash functions, such as message digest 5 (MD5) and Secure Hash Algorithm 1 (SHA1). When HMAC is used with MD5, it is called Hashed Message Authentication Code-Message Digest 5 (HMAC-MD5), and when it is used with SHA1, it is called Hashed Message Authentication Code-Secure Hash Algorithm 1 (HMAC-SHA1) [7, 10, 11]. In our dissertation, we use the secure hash algorithm SHA256 with pre-shared key to form HMAC-SHA256, where the secure hash algorithm SHA256 takes a message of 512-bit blocks as input and returns a digest message with 256 bits as output [7].

2.4.3 Diffie-Hellman (DH) key exchange algorithm

Key exchange algorithms are cryptographic methods that generate cryptographic shared keys that are shared among users. After Alice and Bob agree on a shared key, they can use it in HMAC, and they can also use it in symmetric encryption algorithms to encrypt or to decrypt files. Alice encrypts file using one of the symmetric algorithms and sends it to Bob. Bob in turn uses the same symmetric algorithm to decrypt the file. Note that Alice and Bob must agree on a shared key before using symmetric algorithm. Many key agreement protocols have been proposed. The Diffie-Hellman (DH) algorithm [12] is a very popular example that introduces a key exchange protocol using the discrete logarithm problem [13]. DH algorithm enables Alice and Bob to exchange secure keys over an insecure channel.

Figure 2 shows the mechanism of DH. The values g and p are public parameters known to Alice and Bob, whereby p is a prime number and g (generator of p) is an integer less than p. This means that for all every number n between 1 and p-1, there is a power k of g such that $n = g^k \bmod p$. Both Alice and Bob choose a secret random integer number, a and b, respectively. After that, Alice sends to Bob ($g^b \bmod p$), and Bob sends to Alice ($g^b \bmod p$). Finally, they agree on a secret key by using this formula ($(g^a)^b \bmod p$).

Figure 3 shows that a "man-in-the-middle" (Mallory) can listen and modify the conversation messages between Alice and Bob. In so doing, she can convince Alice and Bob that they are communicating with each other while in fact both are communicating with Mallory [7]. Moreover, **Figure 3** shows that the main vulnerability in the DH protocol is that it does not have an authentication process. Several

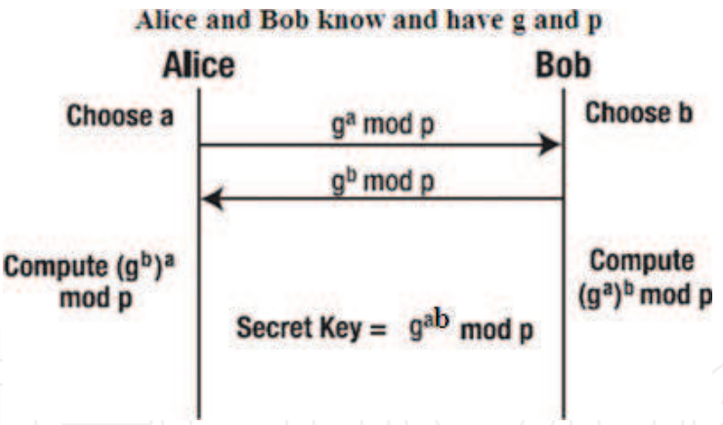

Figure 2.
Diffie-Hellman key exchange algorithm. Alice and Bob agree on a secret key over an insecure channel. The secret key that they agree on is computed as (ga)b mod p.

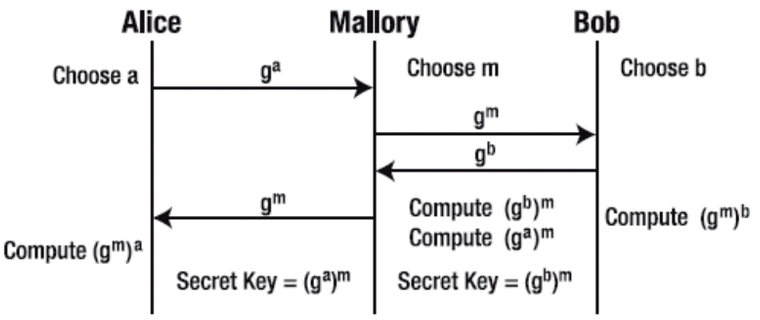

Figure 3.
A Diffie-Hellman weakness. A man-in-the-middle (e.g., Mallory) impersonates Alice to agree on a shared key with Bob. Also, she impersonates Bob to agree on a shared key with Alice.

versions of DH protocol exist to overcome this problem, for example, by using DH with digital signature [11]. Diffie et al. [14] enhance a Diffie-Hellman protocol with an authentication process, whereby Alice and Bob must authenticate themselves using a digital signature. Alice and Bob must have a pair of keys (public key and private key) and a certificate for the public key. So, during execution of DH protocol, Alice and Bob transmit massages with signature; Mallory cannot forge the signature because she needs to share Alice's private key and Bob's private key.

The DH algorithm relies on heavy computation, which may not be suitable to resource-constrained platforms.

3. Related work

There are many researches that target covert channel undetectability [15–17], but most of the works have drawbacks and lack in channel detectability (Girling [18] in 1987). He creates three covert channels through a local area network (LAN): two of them are storage channels, and the third is a timing channel. The two storage channels depend on "what-is-sent" strategy, whereby one of them depends on the frame size, which is sent by the sender. If the frame size equals to 256, the amount of covert information decoded by receiver, who monitors the sender activity on the LAN, would be 8 bits. On the other hand, the timing channel depends on the time that represents the time interval between successive sends. The time difference between successive sends may be odd or even, and the prior agreement between the sender and receiver (who monitors the time between successive sends) is such

that the odd time means bit "0" and even time means bit "1." So, the timing channel obviously depends on "when-is-sent" strategy.

TCP/IP protocol is used to create covert channel that is targeted by many researchers, where they used TCP to hide information [19–22]. Zhang et al. [6] propose covert channel to transfer messages to control (increasing or decreasing) the period of silence in traffic of VoLTE traffic. Create covert channel through hiding information in IP fields [20, 23]. Mead et al. [24] propose timing covert channel for wireless communication; they developed android application to communicate through local area network and mobile network. The results show that the channel is very undetectable in spite of the existence of malware and intrusion detection system.

Some researches, Fatayer et al. [15–17], try to use covert channel as benign channel, and it can be used to send legal information between the client and server. They used gaps in memory to create covert channel. Also, they used the channel to send text and audio files in acceptable time. The proposed technique depends on pre-agreement database which consists of original keys and its corresponding fake keys. Each original key has multiple fake keys. The database consist of the characteristics of clients; each feature represents an original key, and it has multiple fake keys. **Figure 4** figures out the pre-agreement between the client and server before using covert channel.

Customer asks cloud provider to access his resources. The summarization of approach is as follows. First, pre-agreement between the server and client is shown in **Figure 4**. Second, the customer sends a packet which contains "Fakei" attribute belongs to a specific customer (e.g., name) to the cloud provider. Third, the cloud provider will analyze the packet and make sure that the "Fakei" belongs to which customer. If yes, the provider goes to next step. Fourth, the cloud provider will ask for extra information to verify the customer and then he sends a packet that contains another fake key to the customer. Fifth, the customer receives the packet and he verifies the packet. The customer will send the required information to cloud provider such as one of the fake keys of email. Seventh, the cloud provider will analyze the packet to make sure that the "Fakei" (email) belongs to which customer. If yes, the cloud provider will accept the request. Eight, steps from 4 to 7 are considered the first level of security, so if these steps are repeated more than one time, it can achieve multilevel of security.

A new detection approach of covert timing channel is proposed by Fahimeh et al. [25], where this approach enables to detect covet time channel through traffic distribution. They used statistical test to measure the network traffic online.

Characteristics/ fake keys	Original	Fake $_{i=0}$	…	Fake $_{i=n}$
Name	tamer	00001111	…	000000001
id	93/98	……	… …	…
Email	……	……	… …	….
……	…..	….		

Figure 4.
Database as pre-agreement between the client and server.

4. Covert channel with authentication leads to secure communication

The proposed technique is responsible for creating secure communication channel using covert channel, encryption, and authentication. In the first step before using covert channel, the client and server must agree on pre-agreement table as shown in **Figure 4**, where it depicts out the pre-agreement table consisting of original key (OK) and its corresponding fake keys. The key point here is that the fake key is used in communication channel and the original key is kept secret in both sides (client and server). **Figure 5** illustrates the proposed technique, where the network consists of the client and server. The client agrees on shared information (e.g., table as database) with the server.

The client wants to send to the server a message. Then, he encrypts the message using the original key. After that the encrypted message is attached with the fake key to be a parameter to HMAC function. HMAC depends on shared key between the client and server. Then the client sends (HMAC + encrypted message + fake key) to the server. Where, HMAC is used for integrity and fake key for server to know the

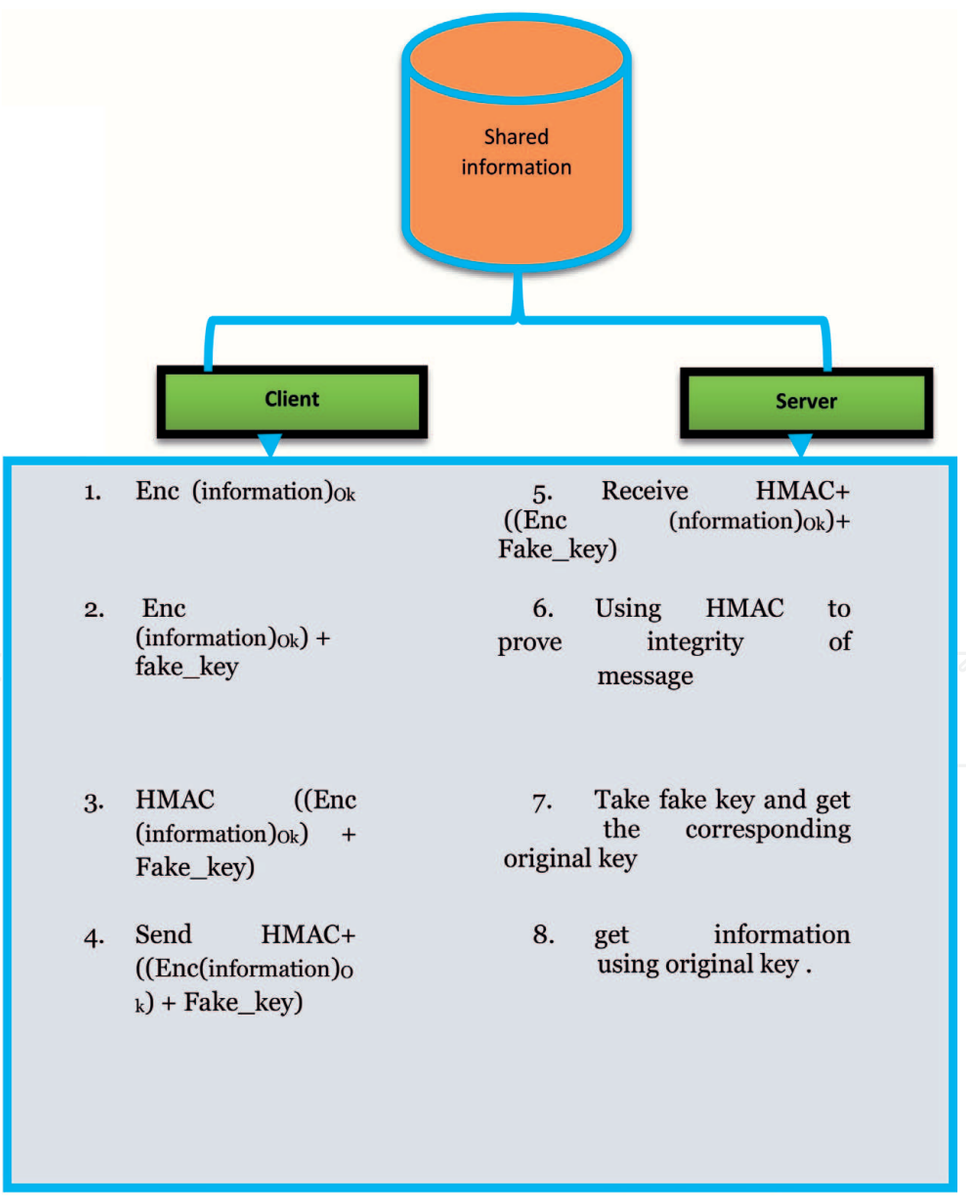

Figure 5.
Technique for transferring secure data through covert channel using covert channel, encryption, and authentication.

original key to decrypt the encrypted message. On the other hand, the server receives the client's message. After that he separates the message to the HMAC, encrypted message, and fake key. Then, he checks the integrity of the message. After that he gets the original key from its corresponding fake key to decrypt the encrypted message.

5. Performance discussion

The implementation of technique is done by: first create a network as client and server with implemented java application. Client and server machines are 32 bit ×86, CPU Core (TM) i5 2.40 GHz, and Ram 4GB. Advanced Encryption Standard (AES) [26] algorithm is used in the implementation. Hashed message authentication code (HMAC) is used to guarantee the integrity [7, 11, 12]. The following issues are satisfying in this technique:

1. Confidentiality: the technique guarantees that the messages are protected from disclosure, which is done by encrypting the messages with original keys that do not send through communication channel. Instead, the original keys are sent encrypted by fake keys.

2. Integrity: the information is protected from being changed by unauthorized parties through using HMAC function which checks if the content of the message is altered or not.

3. Undetectability: undetectability is achieved depending on two conditions: first, plausibility, the messages that are sent through the covert channel are protected from adversary by making the covert channel appear like a normal channel through sending normal encrypted messages, and, second, hiding the fake key inside the message which does not affect bit distribution, especially when the size of the fake key is small.

4. Comparative analysis: My technique is used in two ways, malicious and benign usages. Also, encryption and authentication are used that differ from other techniques.

5. Dynamically: the generating keys between the client and server have a flexible length. Because when you repeat the scenario in **Figure 5** several times, you get new keys with different sizes. If you repeat the technique four times and each time generates key with 16-bit size, then you will get a key with 64-bit size.

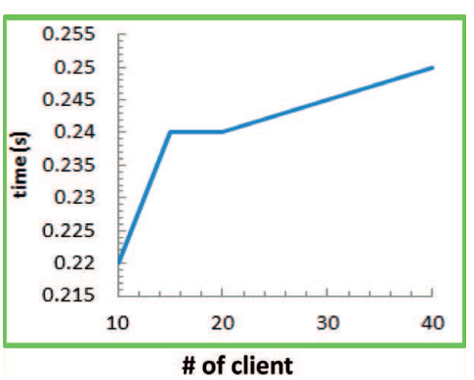

Figure 6.
The technique can handle multiple users in acceptable time.

The time that is needed by the server to serve the client is measured. The technique can handle several clients with acceptable time as shown in **Figure 6**. The server needs less than 0.25 s to deal and handle 30 clients.

6. Conclusion and future work

Encryption is used to achieve confidentiality to protect data from stealing from the third party (e.g., attacker). If users use encryption, they cannot achieve integrity. They need authentication and covert channel besides encryption technique to achieve integrity. In this chapter, we used encryption, authentication, and covert channel to produce a secure communication between eligible parties. This technique satisfies confidentiality through encryption and integrity using authentication algorithm. Finally, this technique generates undetectability covert channel through normal distribution of bits. Secure communication channel between the client and server that enables them to transfer data securely and to agree on keys that are used for future communication. The technique needs pre-shared table that consists of original and fake keys.

Acknowledgements

This is to happily express my sincere thanks and appreciation to the following for their support and guidance throughout the chapter writing. I would like to thank my friends who stood beside me and helped me pursue my work. This chapter is dedicated to my parents, without whom, after the blessings of Allah, all this work would not be possible. They have been a source of endless love, encouragement, and support. They believed in me and in whatever decision I took and are proud of me on whatever achievement I may have.

Thanks

Your support means a lot of thanks.

Author details

Tamer S.A. Fatayer
Computer Science and Information Technology, Al-Aqsa University, Gaza, Palestine

*Address all correspondence to: ts.fatayer@alaqsa.edu.ps

References

[1] Dlaminia EM, Eloffb M. Information security: The moving target. Computers & Security. 2009;**28**:10

[2] Tilborg H, Jajodia S. Encyclopedia of Cryptography and Security. 2nd edition. Boston: Springer; 2011. ISBN: 978-1441959058

[3] Lampson B. A note on the confinement problem. Communications of the ACM. 1973;**16**:613-615

[4] Ray B, Mishra S. A protocol for building secure and reliable covert channel. In: PST'08: Proceedings of the Sixth Annual Conference on Privacy, Security and Trust; Washington, DC, USA; 2008. pp. 246-253

[5] Giffin J, Greenstadt R, Litwack P, Tibbetts R. Covert messaging through TCP timestamps. Presented at the PET'02: The Workshop on Privacy Enhancing Technologies; San Francisco, CA, USA; 2002

[6] Zhang X, Tan Y-A, Liang C, Li U, Li J. A covert channel over VoLTE via adjusting silence periods. IEEE Access. February 2018;**6**:9292-9302

[7] Daswani N, Kern C, Kesavan A. Foundations of Security: What Every Orogrammer Needs to Know. 1st ed. Berkely, USA: Apress; 2007

[8] Sandhu R, Samarati P. Authentication, access control, and audit. ACM Computing Surveys. 1996;**28**:241-243

[9] Brainard J, Juels A, Rivest R. Fourth factor authentication: Somebody you know. In: CCS'06: Proceedings of the 13th ACM conference on Computer and communications security; Virginia, USA; 2006. pp. 168-178

[10] Kumar M. An enhanced remote user authentication scheme with smart card. International Journal of Network Security. 2010;**10**:175-184

[11] Information Technology Laboratory. The keyed-hash message authentication code (HMAC). National Institute of Standards and Technology, Technical Report FIPS PUB 198; 2002

[12] Bellare M, Canetti R, Krawczyk H. Keying hash functions for message authentication. In: CRYPTO'96: Proceedings of the 16th Annual International Cryptology conference on Advances in Cryptology; London, UK; 1996. pp. 1-15

[13] Diffie W, Hellman M. New directions in cryptography. IEEE Transactions on Information Theory. 1976;**IT-22**:644-654

[14] Mahalanobis A. Diffie-Hellman key exchange protocol, its generalization and nilpotent groups [Ph.D. dissertation]. Florida: The Charles E. Schmidt College of Science, Florida Atlantic University; August 2005

[15] Fatayer T, Khattab S, Omara F. A key-exchange protocol based on the stack-overflow software vulnerability. In: ISCC'10: IEEE Symposium on Computers and Communications; Riccione, Italy; June 2010. pp. 411-416

[16] Fatayer T, Khattab S, Omara F. OverCovert: Using stack overflow software vulnerability to create a covert channel. In: NTMS'11: 4th IFIP International Conference on New Technologies, Mobility and Security; Paris, France; February 2011

[17] Fatayer T, Timraz K. MLSCPC: Multi-level security using covert channel to achieve privacy through cloud computing. Presented at the WSCNIS'2015 the 2nd World Symposium on Computer Networks

and Information Security; Hammamet, Tunisia; 2015

[18] Girling CG. Covert channels in LAN's. IEEE Transactions on Software Engineering. 1987;**13**:292-296

[19] Gligor V. A Guide to Understanding Covert Channel Analysis of Trusted System. National Computer Security Center (NCSC) Technical Report, Version 1; 1993

[20] Valentin B, Annessi R, Tanja Z. Decision tree rule induction for detecting covert timing channels in TCP/IP traffic. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction; 2017

[21] Katzenbeisser S, Petitcolas F. Information Hiding Techniques for Steganography and Digital Watermarking. 1st ed. Norwood, USA: Artech House, Inc; 2000. ISBN: 1580530354

[22] Ahsan K. Covert channel analysis and data hiding in TCP/IP [Master of applied Science thesis]. Electrical and Computer Engineering, Toronto University; 2002

[23] Rowland C. Covert channels in the TCP/IP protocol suite. First Monday Journal. 1997;**2**:5

[24] Mead FC, Zielinski JM, Watkins L, Robinson WH. A mobile two-way wireless covert timing channel suitable for peer-to-peer malware. In: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC); Montreal, QC; 2017. pp. 1-6

[25] Rezaei F, Hempel M, Sharif H. Towards a reliable detection of covert timing channels over real-time network traffic. IEEE Transactions on Dependable and Secure Computing. 2017;**4**:249-264

[26] Shao F, Chang Z, Zhang Y. AES encryption algorithm based on the high performance computing of GPU. In: 2010 Second International Conference on Communication Software and Networks; 2010. pp. 588-590