Man in The Middle Attacks Against SSL/TLS: Mitigation and Defeat

Muneer Alwazzeh*, Sameer Karaman and Mohammad Nur Shamma

Electrical and Mechanical Engineering Faculty, Damascus University, Damascus, Syrian Arab Republic
E-mail: malwazzeh@aec.org.sy; samir61mk@gmail.com; shamman01@yahoo.com
**Corresponding Author*

Received 03 February 2019; Accepted 19 April 2020;
Publication 26 June 2020

Abstract

Network security and related issues have been discussed thoroughly in this paper, especially at transport layer security network protocol, which concern with confidentiality, integrity, availability, authentication, and accountability. To mitigate and defeat Man-in-the-middle-attacks, we have proposed a new model which consists of sender and receiver systems and utilizes a combination of blowfish (BF) and Advanced Encryption Standard (AES) algorithms, symmetric key agreement to distribute public keys, Elliptic Curve Cryptography (ECC) to create secret key, and then Diffe Hellman (DH) for key exchange. Both SHA-256 hashing and Elliptic Curve Digital Signature Algorithm (ECDSA) have been applied for integrity, and authentication, respectively.

Keywords: SSL/TLS, MITM, DDoS, integrity, accountability.

1 Introduction

Technology has migrated the traditional shopping to internet-based machines like personal computers, laptops as well as hand held devices like smart

Journal of Cyber Security and Mobility, Vol. 9_3, 449–468.
doi: 10.13052/jcsm2245-1439.933

phones. E-Commerce applications work over client server phenomena, where customer is the client and consumer is the server. The growing number of attacks promotes the development of network security as very critical issue for researchers, organizations, academics and industry. Knowledge of the attacking methods allows development of appropriate security models. Although secure socket layer/transport layer security (SSL/TLS) is the most secure web security protocol [1], it has lots of vulnerabilities resulting from weak cipher support, poor negotiation, weak authentication and integration and misconfiguration, like exploits TLS's cipher block chaining (Lucky 13 Attack), or exploits the HTTP compression technique (Breach Attack) and need quick solutions [2]. Transport layer undergoes many types of attacks, (Eavesdropping attacks, Port scan attack, Reply attack, Man-in-the-Middle attack, Denial-of-Service attack, and so on) [3].

In the beginning, for a better understanding of the subject, we will briefly explain some of the issues related to the transmission of information between the network elements and the protocols adopted in the transport and talk about the security requirements of the network and then summarize a simple explanation of the attacks of man in the middle which is one of the most common attack [4].

1.1 Open system interconnection (OSI) reference model

Consists of seven layers as shown in Figure 1 [5]:

- **The application layer:** supplies the interface of the communication system.
- **The presentation layer:** treats with the composition of data when it moves from one communicating application to another.
- **The session layer:** allows two applications to synchronize their communications and exchange data.
- **The transport layer:** transfers data between two session layer entities.
- **The network layer:** supplies addressing to use it in internet work and route data between two systems.
- **The data link layer:** supplies the connection between network layer and physical network, so ensuring reliable flow of data in the network.
- **The physical layer:** provides the mechanical, physical, and electrical interfaces between systems.

1.2 Secure socket layer/Transport layer security (SSL/TLS)

SSL is the most popularly used protocol for transferring data between client and server. SSL is a successor of TLS; it operates between application and

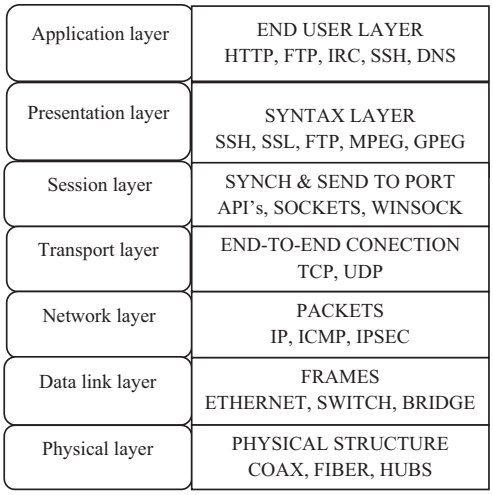


Figure 1 7 layers of the OSI model.

transport layer of OSI reference model. SSL is compatible with Netscape, Microsoft browsers and all other web application products. SSL uses both symmetric and asymmetric encryption techniques for mutual transfer of data between client and server; it also uses digital signatures issued by trusted Certificate Authorities [6].

SSL have three protocols under it: **Handshake protocol**, **Record layer protocol, Alert protocol. Handshake protocol**, which is used to establish the secure connection between client and the server using the cipher suites and other parameters.

Handshake protocol is discussed first as shown in Figure 2.

Step 1: a "client hello" message is sent from client to server that he tries to contact.

Step 2: a "server hello" message is sent from server to the client.

Step 3: a Server Key Exchange message is sent from server to the client.

Step 4: a Server Hello done sent from server after all data have been passed to the client.

Step 5: key information of the client is sent to the server with Client Key Exchange message encrypted with the server public key and only the legitimate server can pass client's information.

Step 6: a Change Cipher Spec message is sent from client to the server to inform that both the parameters of the secured connection and activate are the same.

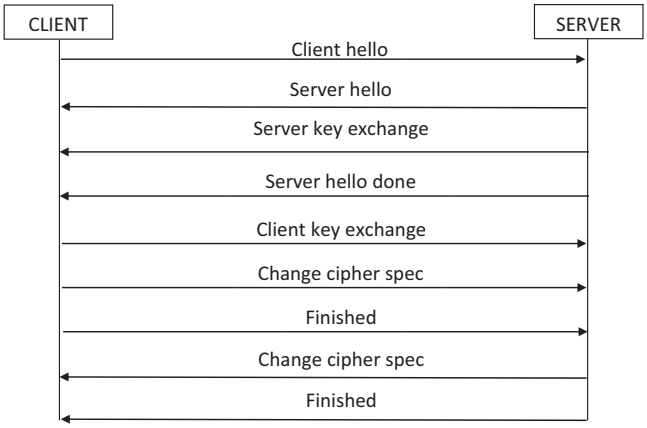


Figure 2 SSL handshake.

Step 7: finished message is sent from the client.

Step 8: the same Change Cipher Spec is sent from the server to the client, notify the options in the secured connections, after that finished message is sent to the client, and verify all the options.

Record layer protocol is used to encrypt the data that is sent through network using the key established during the handshake protocol. This layer is handles the actual data. It gets data from the application layer, encrypts it, fragments it to an appropriate size, as determined by the algorithm, and sends it on to the Transport Layer. Additionally, this layer can optionally compress or decompress data based on if the data is being sent or received.

Alert protocol is used to send the custom messages to others whenever they detect any intrusion in system. It is used to alert status changes to the peer. The primary use of this protocol is to report the cause of failure. Status changes include such things as error condition like invalid message received or message cannot be decrypted, as well as things like the connection has closed [7].

1.3 Basic concepts of SOA security goals

Confidentiality, Integrity, Availability (CIA), Authentication, and Accountability (nonrepudiation) are the major security requirements in information networks [8].

- **Confidentiality:** data should be readable to actors with appropriate permission. Cryptography and access controls are the best method to

protect the confidentiality in the information systems. Intruder's social engineering and malware are example of threats to confidentiality.

- **Integrity:** prevents the improper and unauthorized modification of information. There are two types of integrity protection mechanism: first, is a preventive mechanism, so the prevention of unauthorized modification of information is done by access control, and second is a detective mechanism, which can detect unauthorized modifications if the preventive mechanism has failed.
- **Availability:** only authorized users can access the information. Using intrusion detection system and firewall enables mitigation of attack against availability.
- **Authentication and Authorization:** To make information available to those who need it and who can be trusted with it, organizations use authentication and authorization. Authentication proves that a user is the person he or she claims to be. That proof may involve something the user knows (such as a password), something the user has (such as a "smartcard"), or something about the user that proves the person's identity (such as a fingerprint). Authorization is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity, such as reading a file or running a program.
- **Accountability (nonrepudiation):** before the user carry out the activity that authorized to perform, he must be authenticated. Nonrepudiation means that when the methods of authentication cannot refuted, the user cannot later deny that he or she performed the activity.

1.4 Man-In-The-Middle-attack (MITM)

A man-in-the-middle attack is a type of cyber-attack so the attacker can inserts him/herself between two parties, and can imitate both parties and access information that send to each other. A man-in-the-middle attack permits attacker to intercept, send and receive information meant for someone else, or not meant to be sent at all.

Man in the middle attack interrupts data between server and client secretly. It mainly captures public key of server and its own public key to client, and client assumes that it is server public key and send further information to attacker but not server [9, 10].

A simple MITM attack model is shown in Figure 3.

There are five classification, which identify the vulnerabilities that attackers leverage to implement MITM attacks [4].

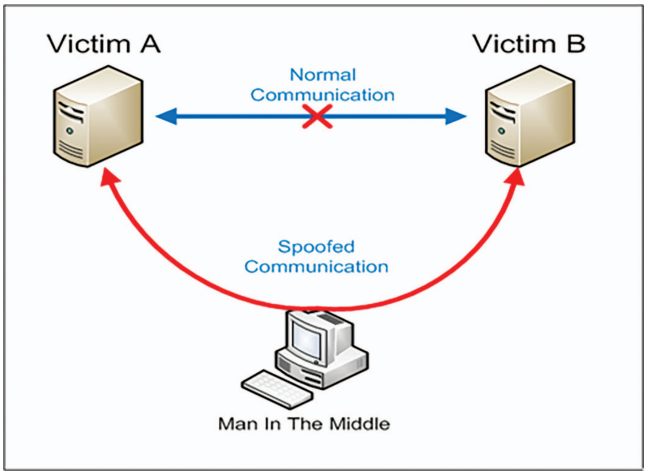


Figure 3 Man in the middle attack model.

(1) **Cipher Block Chaining:**Block ciphers require blocks of fixed length. If data in the last block is not a multiple of the block size, extra space is filled by padding. The server ignores the content of padding. It only checks if padding length is correct and verifies the Message Authentication Code (MAC) of the *plaintext*. That means that the server cannot verify if anyone modified the padding content. Attackers can use inherent vulnerabilities in the Cipher Block Chaining (CBC) mode of operation to decrypt the contents of an HTTPS message. (Example lucky 13 and poodle attack).

(2) **Compression:** A key part of HTTPS communications is the compression of message contents to reduce resource usage. Attackers can exploit message compression by comparing size differences, allowing the inference of message contents. (Example, crime, breach, and time attack).

(3) **Export Key:** This classification applies to attacks that exploit export grade security keys. These keys originally introduced to comply with United States cryptography export regulations. The regulations limited the strength of cryptography software with the intention that the weaker export keys could be broken by United States government agencies. However, attackers are also able to exploit these export grade security keys in order to attack the HTTPS communications and decrypt the contents of the communications. (Example, logiam, and freak attack).

(4) **Implementation Error:** These errors are typically the result of a poorly applied security feature or a bug in the system. Attackers can exploit

these implementation errors to launch attacks. (Example, berserk, komodia redirector, ccs injection, drown, and heartblead attack).

(5) **Renegotiation:** Renegotiation allows HTTPS connection parameters and keys changed in existing connections upon request. Attackers can exploit the Renegotiation feature to make their own connection and then splice another connection to use the attackers' connection settings. (Example, triple handshake attack and renegotiation attack).

2 Related Work

MITM attacks are performed using communication layers. Open System Intercommunication (OSI) and GSM networks are the most affected communication channels by MITM attacks. Preventing MITM attacks requires a few down to earth ventures with respect to clients, and additionally a combination of encryption and check techniques for applications. In order to thwart such attacks, various proposals have emerged. Some proposals focus on enhancing the certificate authentication model. Other proposals focus on strengthening client authentication.

Here is some of the best mechanism to improve authentication and protect against MITM attacks:

1. **Third-Party Solutions:**are the most popular approach that provides a protection of the first connection to a new domain, and scalable attestation of certificates for all public domains and minimal requirements for web applications. Unfortunately, this approach also faces several critical challenges. First, theseapproaches have significant costs. Second, the complexity of the resulting trust model. Third, these mechanisms introduce new privacy risks. Finally, certificate revocation procedures become more complex [11].
2. **Detection and mitigation using prior knowledge:**
   If the client has information prior to connecting to the server, the client may be able to detect MITM. Several approaches to accomplish this exist:
   - DNS (domain name server) certification authority authorization (DNS record type CAA) [12].
   - Mitigates attack on CAs by restricting which CAs that can issue certificates for a host.
   - DNS- based Authentication of Named Entities (DANE) Transport Layer Security (TLS) protocol (DNS record type TLSA) [13].

- Mitigates attacks on server by sending the server certificate to client in the DNS response.
- HTTP Strict Transport Security (HSTS) [14].
- Mitigates attacks on servers by enabling web sites to declare themselves accessible only via HTTPS.
- Public Key Pinning Extension for HTTP [15].
- Enabling web sites to declare fingerprints of allowed server and CA certificates to mitigate attacks on server.

The problems with the above mechanisms is that the DNS- based mechanisms requires deployment and use of DNSSEC, while the HTTP- based mechanism requires that the user recently visited the website, and that no MTIM attacker was present during the first visit.

3. **Direct Validation of Certificates** (DVCert) is an efficient and easy to deploy protocol that provides stronger certificate validation and effective detection of MITM attacks without using third parties. A DVCert transaction uses a modified Password Authenticated Key Exchange (PAKE) protocol known as PAK [16, 17]. DVCERT modified PAK to provide only server authentication and integrity protection instead of mutual authentication and generation of encryption keys (i.e., traditional use of PAKE protocols). These changes allow better performance and simplify deployment without affecting PAK's formal security proofs [11].

3 Background Information

Before we present our proposed model, we will review some important definitions that are at the heart of our proposal and will increase our understanding of the model:

ELLIPTIC CURVES IN TLS: is an attractive public-key cryptosystem used for mobile environments. RFC 4492 [18] specifies the list of elliptic curve systems being used in TLS.

Elliptic Curve Cryptosystem (ECC) offers comparable security to RSA (*Rivest–Shamir–Adleman*) with small key size [19]. ECC requires less computational power, low bandwidth and minimum memory.

Figure 4 shows Elliptic Curve Cryptosystem

Elliptic Curve Diffie-Hellman (ECDH): it is an algorithm used to establish a shared secret between two parties. DH used for exchanging cryptography key in symmetric encryption algorithms like AES. In addition, it supplies

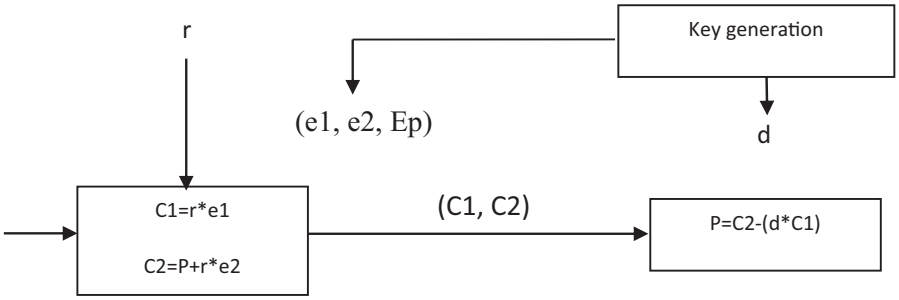


Figure 4 Elliptic curve cryptosystem.

the basis for a variety of authenticated protocols, and uses to supply forward secrecy in Transport Layer Security's ephemeral modes [20].

Forward Secrecy: is the property of secure communication protocol that protect past sessions against future compromises of passwords or secret keys.

Forward secrecy can accomplish by computing a random key for every session. This ensures that if one key compromised, it does not result in the loss of integrity of other keys that may be generated later for the session [21].

Digital Signatures: digital communication is used to assure the authentication of message and electronic documents using various encryption methods to supply unmodified and original documentation. Digital signatures is used in software distribution, financial transactions, and e-commerce, which depend on tampering or forgery detection techniques.

Elliptic Curve Digital Signature Algorithm (ECDSA): ECDSA have a smaller key size, which guides to reduction in processing power, bandwidth and storage space, and faster computation time [22].

Secure Hash Algorithm (SHA): is a set of algorithms to provide better online security standards for organization and the public. SHA-3 as atop – level secure hash algorithm keeps sensitive data safe and prevent different types of attacks [23].

Blowfish Algorithm (BF): BF is a symmetric encryption algorithm that uses a variable-length key block cipher. It is a Feistel network, enables iteration with the encryption function up to 16 times. The size of block is 64 bits and the key lies at any length up to 448 bits [24].

This algorithm divided into two parts as can be shown in Figure 5:

Key expansion part, which converts a 448 bits key into several sub-key arrays of 4168 bytes in total, and Data encryption part performed through

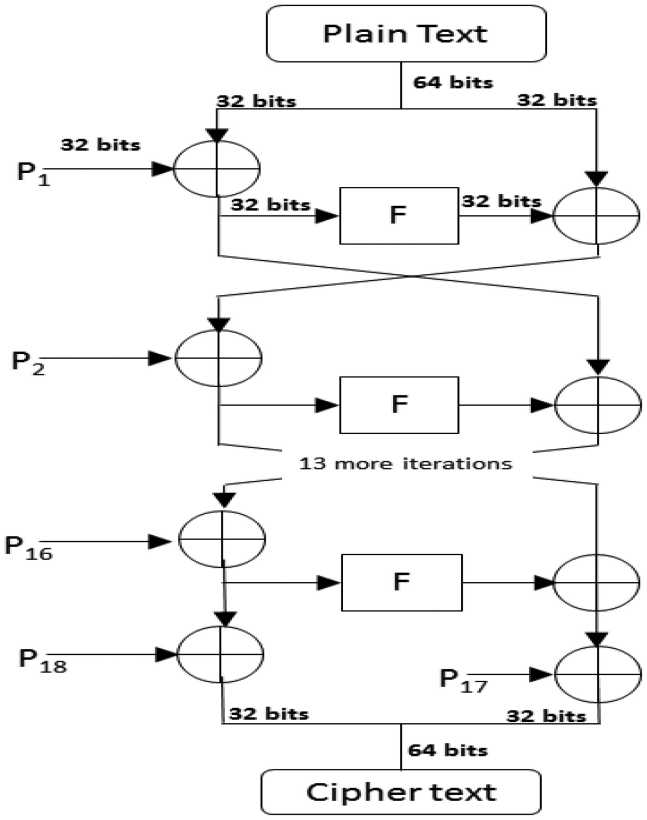


Figure 5 Blowfish algorithm.

16-rounds. Depending on key and substitution, every round carries out a permutation based on data using XOR operation and addition on 32-bit words.

Blowfish has no known security weaknesses and also Blowfish algorithm gives more throughputs as compared to other symmetric encryption algorithms [25], and it in available freely for anyone. Therefore, this is the reason of its popularity in encryption algorithms.

Advanced Encryption Standard (AES): AES is a block cipher symmetric algorithm with block length 128 bits and key lengths of 128, 192 or 256. The number of round determines the key size of the algorithm [26].

The algorithm is shown in Figure 6.

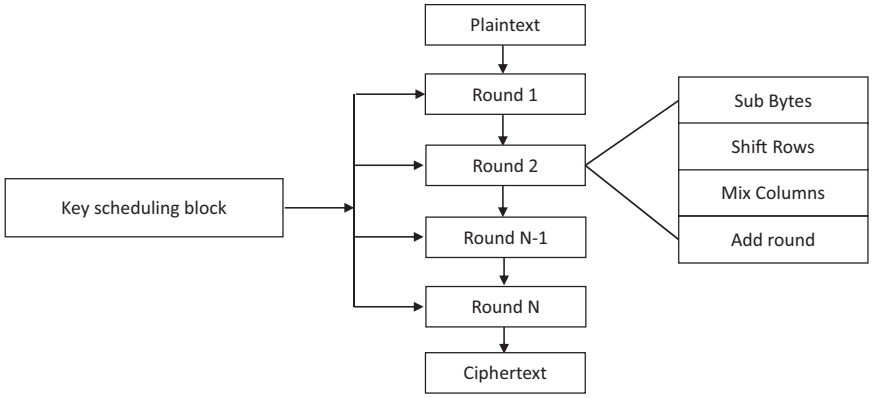


Figure 6 AES algorithm.

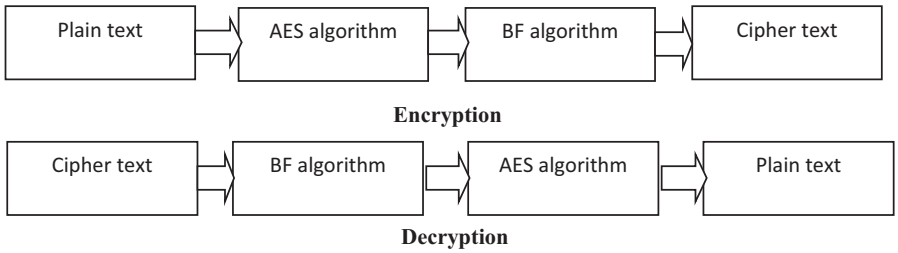


Figure 7 Hybrid AES and BF.

AES algorithm divided into four parts:

(1) Key expansion – It derives the keys for each round.
(2) There is initial round before first round – carry out Add round key.
(3) N-1 Rounds – carry out all four transformations.
(4) Final Round Nth round – in this round, only Mix Columns transformation is missing.

Changing the column of S-Boxes improve AES algorithm.

AES suffer from Brute force attack but it gives more security when it compared with another algorithms. The throughput of AES is less as compare to the Blowfish but when we are more concerned about security, AES is best.

Figure 7 shown a combination of BF and AES algorithms for encryption and decryption.

Key Management

Keys used in cryptography to achieve confidentiality or data integrity. Key management process is the set of processes, which are required to establish the key and distribute the same to authorized parties. The processing of sharing the key between more than one parties, called Key Distribution [27]. The shared key in symmetric key agreement, established between two parties without any Key Distribution Centre (KDC).

Message Authentication/Entity Authentication and Digital Signature

In secure communication, Hashing is very important technique that provides all requirement of data security like integrity, confidentiality, and authentication. Password hashing is lightweight and convenient to use and can defend against phishing attacks.

4 Proposed Model

Network security involves methods or practices used to protect a computer network from unauthorized accesses, misuses or modifications. To improve database security and prevent the tampering of data, we use data encryption methods. Development of encryption algorithm is very important in information security but data security consume a huge resource (battery power, and CPU time).

Cryptography is one of techniques used to secure and guarantee data confidentiality by doing conversion to the plaintext (original message) to cipher text (hidden message) with two important processes, encrypt and decrypt. To mitigate and defeat MITM attacks, our proposed model consists of two sender and receiver system. It depends on combination of two symmetric encryption algorithms, blowfish (BF) and Advanced Encryption Standard (AES) algorithms [28], for increasing the speed of encryption/decryption. Then we will distribute Public keys by the symmetric key agreement protocol, created secret key by Elliptic Curve Cryptography (ECC), and applied to Diffe Hellman (DH) for Key exchange. SHA-256 hashing used for integrity, and for Authentication we used Elliptic Curve Digital Signature Algorithm (ECDSA).

Message integrity achieved by cryptographic hash functions. Hashing algorithms are MD2, MD4, MD5 and SHA-1, SHA-2, SHA-3. SHA is more

Table 1 Features and benefits of our proposed model

Features	Benefits
Using blowfish and AES algorithms	Enhance the speed of data encryption and decryption resolve the problem of key distribution and authentication High computing speed and anti-attack capability
Using Elliptic Curve Diffie-Hellman (ECDH) and Forward Secrecy	Supply a variety of authenticated protocols Secure communication protocols
Using SHA-256 hashing	More secure than MD5
Using Elliptic Curve Digital Signature Algorithm (ECDSA)	Small key size reduce processing power, bandwidth and storage space fast computation time assure the authentication of message

secure than MD5 but on the other hand, MD5 is faster than SHA on 32-bit machines. By doing digital signature, we achieved to security goals like Authentication and non-repudiation.

Hash code known as message authentication code (MAC), and it is a fixed-size fingerprint of a variable-sized message.

AES is more secure than the Blowfish algorithm. Blowfish gives high throughput as compared to others. The hybrid of AES and Blowfish algorithm has characteristics of both the algorithms and it cannot only enhance the speed of data encryption and decryption, but resolve the problem of key distribution and authentication. In addition, it has high computing speed and anti-attack capability, especially Man In The Middle attacks, which is very hard to detect, and as a result improved the security of data transmission process effectively.

Table 1 shows some features and benefits of our proposed model:

For the proposed model, see Figure 8 for sender system and Figure 9 for receiver system.

We have in the proposed model a system for sending files and a system for receiving files. The sending file can be an audio, an image, a video, or a text file. This file will be an input to the sender system. In the proposed model, we used merging of two symmetric algorithms and this will increase run time but will greatly enhance the security and thus the difficulty of penetration.

Sender's system architecture:

Initially the sender system will encrypt the data using AES and BF algorithms, and an initial key is entered by the sender system at the time of

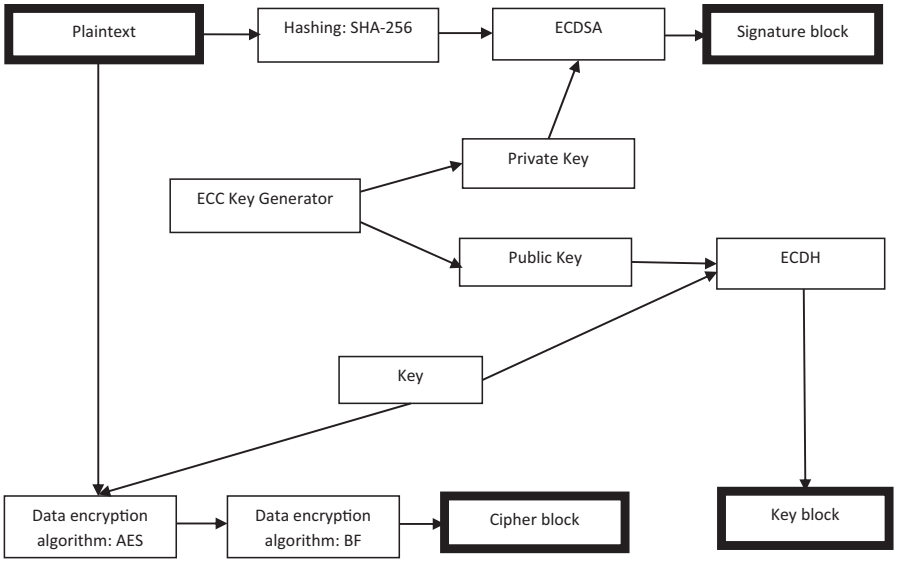


Figure 8 Sender's system architecture.

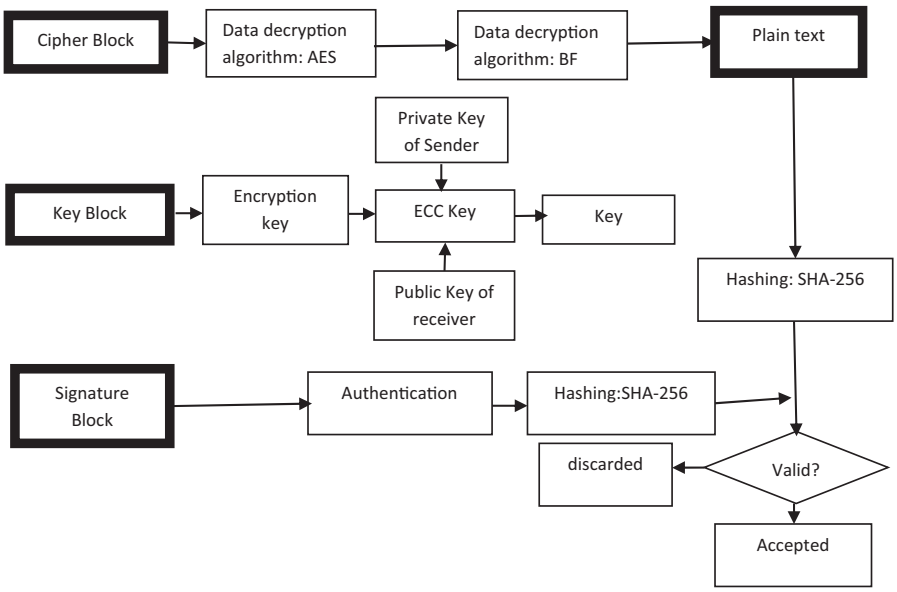


Figure 9 Receiver's system architecture.

encryption. This encrypted key will be encrypted again by ECC concept and sent to the channel by the key management algorithm ECDHA.

First plaintext or other file type will be input of the sender system. Hashing function (SHA-256) has been applied on the plaintext and will give 256-bit message digest value. At the same time, ECC generator will generate a private key and a public key. AES and BF Algorithms are applied on plain text using a key to generate ciphertext. The digital signature is applied on the Hashed result by using a private key and this will generate a signature block. Apply ECC encryption on AES key using public key will give AES key block. We will apply digital signature on encrypted file, and then encrypted file will be sent along with encrypted key to destination. Now sender system will send encrypted file consisting of three blocks: encrypted data (Cipher block), encrypted key (Key block) and signature block.

Figure 8 shown Sender's system architecture.

Receiver's system architecture:

The receiver system receives the encrypted file and decrypts the received encrypted data using the decryption algorithms AES and BF. Applying decryption of ECC algorithm on the encrypted key will give the private key of sender and public key of receiver. Applying the private key of receiver on AES key block will generate AES key. Applying AES key on cipher block will generate plaintext and by hashing function will give a reference result (256-bit message digest). Applying the public key on signature block for authentication will generate a reference result (256-bit message digest). The two outputs (reference results) are compared for the validation process. If both message digests are identical, the data will be accepted; otherwise the data will be discarded. Figure 9 illustrates the structure of the receiver system

5 Conclusion and Future Work

MITM is an active attack, and very difficult to detect, mitigate, and defeat. We should use strong mutual authentication techniques, encryption and decryption algorithm, proper configuration of client and server handshake mechanism to reduce this attack.

Denial Of Service attack is another type of attack at transport layer, try to overcome server resources and network. In future, we will propose a new model to detect and prevent this attack.

Acknowledgments

Prof. Hasan Abou Alnoor has contributed to the work presented here but unfortunately passed away just before submitting it. Authors sincerely dedicate this work to the memory of Prof. Abou Alnoor.

References

[1] K. Bhargavan, C. Fournet, M. Kohlweiss, A. Pironti, P. Strub, Implementing TLS with Verified Cryptographic Security, 2013 IEEE Symposium on Security and Privacy, 2013, pp. 445–459.

[2] A. Satapathy, L.M.J. Livingston, A Comprehensive Survey on SSL/ TLS and their Vulnerabilities, International Journal of Computer Applications, 153 (2016) 31–38.

[3] H. Parmar, A. Gosai, Analysis and Study of Network Security at Transport Layer, International Journal of Computer Applications, 121 (2015) 35–40.

[4] S. Stricot-Tarboton, S. Chaisiri, R.K.L. Ko, Taxonomy of man-in-the-middle attacks on HTTPS, TrustCom 2016, IEEE Computer Society, Tianjin, China, 2016, pp. 527–534.

[5] A. Madan, A. Tuteja, Bharti, OSI Reference Model, International Journal of Advanced Research in Computer Science and Software Engineering, 4 (2014) 55–49.

[6] T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, URL https://www.ietf.org/rfc/rfc5246.txt, IETF, 2008.

[7] T. Shubh, S. Sharma, Man-In-The-Middle-Attack Prevention Using HTTPS and SSL, International Journal of Computer Science and Mobile Computing, 5 (2016) 569–579.

[8] A. Singh, A. Vaish, P.K. Keserwani, Information Security: Components and Techniques, International Journal of Advanced Research in Computer Science and Software Engineering, 4 (2014).

[9] P.K. Pateriya, S.S. Kumar, Analysis on Man in the Middle Attack on SSL, International Journal of Computer Applications, 45 (2012) 43–46.

[10] Radhika, P., Ramya, G., Sadhana, K., Salini, R., Defending Man In The Middle Attacks, International Research Journal of Engineering and Technology, 4 (2017) 579–585.

[11] I. Dacosta, M. Ahamad, P. Traynor, Trust No One Else: Detecting MITM Attacks Against SSL/TLSWithout Third-Parties, Converging

Infrastructure Security (CISEC) Laboratory, Georgia Tech Information Security Center (GTISC), Georgia 2013.

[12] P. Hallam-Baker, R. Stradling, DNS Certification Authority Authorization (CAA) Resource Record, URL http://tools.ietf.org/html/rfc6844, IETF, 2013.

[13] P. Hoffman, J. Schlyter, The DNS Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, URL http://tools.ietf.org/html/rfc6698, IETF, 2012.

[14] J. Hodges, C. Jackson, A. Barth, HTTP Strict Transport Security (HSTS), URL https://tools.ietf.org/html/rfc6797, IETF, 2012.

[15] C. Evans, C. Palmer, R. Sleevi, Public Key Pinning Extension for HTTP, URL https://tools.ietf.org/html/rfc7469, IETF, 2015.

[16] V. Boyko, P. MacKenzie, S. Patel, Provably Secure Password-Authenticated Key Exchange using Diffie-Hellman, in: B. Preneel (Ed.) International Conference on the Theory and Application of Cryptographic Techniques , May 14–18, 2000 Springer, Bruges, Belgium, 2000, pp. 156–171.

[17] P. MacKenzie, The PAK suite: Protocols for Password-Authenticated Key Exchange, DIMACS Technical Reports, Bell Laboratories, Luent Technologies, Murray Hill, USA, 2002.

[18] S.B. Wilson, N. Bolyard, V. Gupta, C. Hawk, B. Moeller, Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS), URL https://tools.ietf.org/html/rfc4492, IETF, 2006.

[19] M.J.B. Robshaw, Y.L. Yin, Elliptic Curve Cryptosystems, An RSA Laboratories Technical Note, URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.461.1411&rep=rep1&type=pdf, RSA Laboratories, 1997.

[20] P. Sehgal, N. Agarwal, S. Dutta, P.M.D.R. Vincent, Modification of Diffie-Hellman Algorithm to Provide More Secure Key Exchange, International Journal of Engineering and Technology, 5 (2013) 2498–2501.

[21] Y. Sheffer, R. Holz, P. Saint-Andre, Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), URL https://tools.ietf.org/html/rfc7525, IETF, 2015.

[22] G. Sarath, D.C. Jinwala, S. Patel, A survey on elliptic curve digital signature algorithm and its variants, Computer Science & Information Technology, 4 (2014) 121–136.

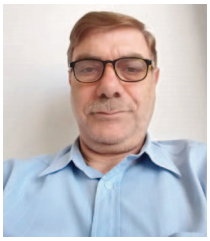
[23] N. Sklavos, Towards to SHA-3 Hashing Standard for Secure Communications: On the Hardware Evaluation Development, IEEE Latin America Transactions, 10 (2012) 1433–1434.

[24] T. Nie, C. Song, X. Zhi, Performance Evaluation of DES and Blowfish Algorithms, Biomedical Engineering and computer Science International Conference, IEEE, 2010.
[25] A. Nadeem, M.Y. Javed, A Performance Comparison of Data Encryption Algorithms, 2005 International Conference on Information and Communication Technologies, 2005, pp. 84–89.
[26] S. Rehman, S.Q. Hussain, W.G.a. Israr, Characterization of Advanced Encryption Standard (AES) for Textual and Image data, International Journal Of Engineering And Computer, 5 (2016) 18346–18349.
[27] A. Menezes, P.v. Oorschot, S. Vanstone, Key Management Techniques, CRC Press, 1996.
[28] A. Mahmud H, B. Angga W, Tommy, A. Marwan E, R. Siregar, Performance analysis of AES-Blowfish hybrid algorithm for security of patient medical record data, Journal of Physics: Conference Series, 1007 (2018) 012018.

Biographies

Muneer Alwazzeh is a Ph.D student at the University of Damascus since summer 2016. He attended the University of Damascus, Syria where he received his B.Sc. in Electrical Engineering in 2000. Muneer has gained an M.Sc. in programming and operating systems from the University of Damascus, Syria in 2010. He is currently completing a doctorate degree in Computer Engineering and Networks at the University of Damascus. His Ph.D. work concentrates on reaching an adaptive and scalable security solution with time and cost optimization that contributes to protecting the computer network and helping traditional programs and tools to protect the network and combat cyber-crime.

Sameer Karaman is an academic staff of electrical and mechanical engineering faculty, Damascus University since 1994. He has also been appointed as academic staff of Private International Syrian University, Virtual University of Damascus, Qasuoun Private University of science and technology, Yarmook Private University, and Al-Rasheed Private University in the period 2008–2020. He was chosen as the head of division of computer engineering and control during 2015–2019. His work interest is in the field of encryption and information security.



Mohammad Nur Shamma is an academic staff of electrical and mechanical engineering faculty, Damascus University since 1994. He has also been involved as academic staff in both the virtual University of Damascus – Information Technology since 2013, and communication and information engineering faculty of Arab International University since 2015. His work interest is in the field of arithmetic science and encryption and information security.