Transport Layer Security Protocol for Intranet

Mohammed Adeeb AbdulJabbar
Department of Computer Science
University of Anbar
Ramadi, Iraq

Ali Makki Sagheer
Information System Department
College of Computer
University of Anbar
Ramadi, Iraq

Ayoob Abdulmonem Abdulhameed
Department of Computer Science
University of Anbar
Ramadi, Iraq

ABSTRACT
Key management is the hardest part of cryptography. Designing secure cryptographic algorithms and protocols isn't easy. As the Intranet becomes popular, it is important to consider the system security. This is because the data flowing through the network is susceptible to be intercepted and modified by a cracker or hacker. So, how to protect personal privacy and preserve a safe online commerce? These are challenges for security protocols. In this paper, a protocol has been developed that depends on the Elliptic key cryptosystem to provide a robust mechanism for key exchange. Also the confidentiality is provided using AES and RC4 with random selection. To satisfy message integrity, SHA1 technique is considered.

General Terms
Security, Networks.

Keywords
Security protocol, Transport layer, Intranet, Key exchange.

1. INTRODUCTION
Security systems typically consist of a number of terminals such as people, computers or other devices, which are communicating through a variety of channels. Security protocols represent the rules by which these communications are governed. Protocols are typically designed in order to avoid any attack or malicious act as possible as. Protection against all possible threats is too expensive; therefore protocols are designed under certain assumptions about the attacks. They may be extremely simple or very complex.

A protocol that incorporates security objective is called a security protocol. Security protocols shall particularly provide security properties of distributed systems. Cryptographic protocols are security protocols that use cryptographic techniques such as encryption methods and digital signature algorithms as basic components [1].

Network security protocols provide a mechanism to securely communicate over public networks, such as the internet, personal and business interactions, facilitating electronic commerce and transactions that require some level of security. The aim of a security protocol is to provide the required combination of the general security services, such as Authentication, Confidentiality, Integrity or Non-repudiation [2].

Protocols that provide secure communication channel over an untrusted network are considered one of the most important parts of today's computing infrastructure. Examples of such common protocols are SSL [3], TLS [4], Kerberos [5], IPSec [6] and IEEE 802.11i [7] protocol suites. SSL and TLS are widely used by web servers and internet browsers to secure the transactions in applications like online banking and other e-commerce applications. The IPSec protocol suite is widely used to provide confidentiality and integrity services over the IP layer and it is commonly used to secure corporate VPNs. IEEE 802.11i offers data protection and message integrity for wireless local area networks, while Kerberos is used for network authentication.

2. RELATED WORKS
In 2003, Wooseok Ham proposed two secure and efficient E-commerce protocols: mobile payment system and on-line sealed-bid auction. These two protocols were built based on number-theoretic hard problems such as DLP and used digital signature and cryptographic hash function as major primitives [8].

Gon Kim, in December 2004, introduced various types of security protocols and addressed general attack types on them. He proposed an ACG-C# tool that can be used to automatically generate C# implementation code for the security protocol verified with Casper and FDR (Failure and Divergence Refinement). With this tool, the security weakness of security protocols which may occur in the implementation step are reduced [9].

Anupam Datta, in September 2005, conducted a study on security analysis of network protocols for his PhD degree. The study addressed two major problems associated with the design and security analysis of network protocols that implement cryptographic primitives. The first problem is related to the secure composition of protocols, which means that to prove properties of complex protocols, the goal is to develop methods for combining independent proofs of their parts. The second one pertains to the computational soundness of the symbolic protocol analysis. This means that, at a high-level, a logical method for protocol analysis must have an associated soundness theorem. This should guarantee that a completely symbolic analysis or proof has an interpretation of the standard complexity-theoretic model of modern cryptography [10].

In August 2012, Sukalp Bhople performed a number of experiments to analyze the DoS vulnerabilities in SSL/TLS Protocols. His experiment included a study of the SSL protocols so as to find a number of SSL functionalities that are likely to be the weak-link and can be used to perform the DoS attacks. He also reviewed the implementation of Openssl to investigate the existence of DoS attack vulnerabilities in the implementation. The experimental results of the study showed that the client authentication can create a significant computational overhead on the server side [11].

Also, there are many examples in the literature of protocols that have been widely used before it turns out that an attack can be taken against the protocol, even though the protocol received intensive analysis, and thought to be correct before they were found to be flawn.

For instance, the Needham-Shroeder authenticated key distribution protocol [12] which was found to allow an intruder to pass an old compromised session key as a new one to a legitimate party [13]; Burrows, Abadi, and Needham [14] showed that, for a protocol in an early draft of the CCITT X.509 draft standard [15], an intruder can cause an old session key whether or not it had been compromised to be accepted as a new one; Also, Pereira and Quisquater [16] showed that two attacks on the Group Diffie-Hellman protocol [17] can be taken; another example is an attack on the Internet Key Exchange (IKE) protocol was independently found by Zhou [18] and Ferguson and Schneier [19]. Kats and Shin [20] addressed the case of attacks by malicious insiders for authenticated key exchange protocols. Pereira and Quisquater [21] suggested a systematic method to derive an attack against any Authenticated Group Diffie-Hellman (A-GDH) type protocol with at least four participants and exhibit protocols with two and three participants.

3. NETWORK SECURITY PROTOCOLS

More and more human interaction these days are taking place over networks instead of face-to-face. The rapid growth of network technologies as both individual and business communication channels have created a growing need for security and privacy. This has led to several security protocols and standards. Among these are: Secure Socket Layer (SSL) and Transport Layer Security (TLS) Protocols; secure IP (IPSec); Secure HTTP (S-HTTP), secure E-mail (PGP and S/MIME), DNDSEC, SSH, and many others.

Prior to the development of these protocols, security attacks have also been developing. Attacks can be directed against the cryptographic algorithms used in protocols, against the cryptographic techniques used to implement the algorithms and protocols, or against the protocols themselves[22].

4. DESIGN OF THE PROTOCOL

Security protocols relay the most on key exchange mechanisms because they provide the means to share secret keys. Therefore, the elliptic curve algorithm has been chosen to increase the level of security needed during the key exchange process. The proposed protocol tends to secure organizations transactions over a private network by providing a secure and robust technique to exchange secret keys safely. The protocol was implement using the windows form application of the .Net environment and it was tested on a group of 20 PCs connected through a wireless network. The protocol performs two major operations to secure transactions: Handshake and data transfer. These operations are shown in more details below:

4.1 Handshake

The client and the server make several connections. They both use the elliptic curve technique to exchange messages. When the handshake process is completed, client and server should both have a shared secret key of a predefined encryption algorithm (AES or RS4). Figure 1 shows the handshake protocol.

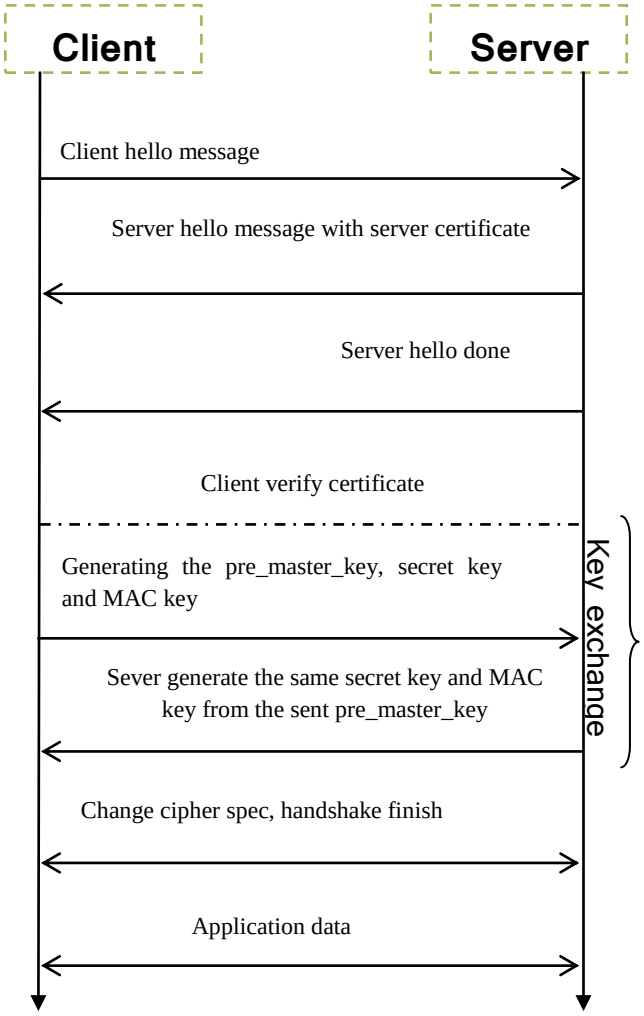

Fig 1: The Handshake Process

1- Client Hello: the client sends hello message that includes a list of cipher suite supported by the client and the requesting the public key of the server. The client also generates a random number.
2- Server Hello: the server sends a response of the client hello message that includes the server public key. The server will choose a cipher suite from the list sent by the client and generate a random number.
3- Server Certificate: the server sends a certificate to the client to verify his identity. This certificate is followed by a server hello done to notify the client with the completion of the Hello procedure.
4- Client creates another random number: pre_master_secret (encrypted with the server's public key using an elliptic curve algorithm) to produce a master_key which will then be used with the two random numbers generated during the Hello step to create the secret key and MAC key.
5- Server decrypts the message that contains the pre_master_key sent by the client and generates the same master key as the client.
6- Change cipher specification: sent by the server and then the client copies the pending cipher spec. into the current cipher spec. After that, the client sends the finished message.

7- At this point, both are ready to transmit data encrypted with the created secret key after sending a handshake "finished" message to each other.

4.2 Data transfer

Both sides should have shared secret key of a specified cipher suite. Now the client and server can communicate by sending encrypted messages with the hash code of the message for data integrity. Each message is encrypted using either AES or RC4 based on the importance and application selection from the previous step. The key length for these algorithms is initially 512 and can be adjusted to become 1024 if a higher level of security needed. The sender encrypts the message and embeds the hash code along with the protocol header to produce the complete packet. At the receiving site, the receiver decodes the packet to get the cipher text and the message hash code, and then compares the computed hash to the received one to ensure that the message hasn't been modified during the transmission.

5. RESULTS AND DISCUSSION

The problem is that the protocols tend to focus on either security or performance. Many protocols have been proposed to deal with the problems of security in both wired and wireless networks. In this paper, a protocol was proposed that deals with the following issues:

1- **Authentication:** establishing session keys between communicating parties is essential when using symmetric cryptographic primitives to protect confidentiality and integrity. Currently most of the applications uses RSA. A new promising alternative is ECC. The following table shows a comparison between the most used key exchange mechanism: the RSA, and the new promising ECC technique.

2- **Confidentiality:** two algorithms have been implemented for this purpose. The first is a block cipher, the AES, with variable key length up to 1024. The second one is a stream cipher, the RC4, which is used when high speed communication is on the board.

AES and RC4 are two encryption ciphers that are used in a variety of applications. A common example where both ciphers are employed is in wireless routers. The following Figure 2 shows the performance of RC4 and AES algorithms in terms of sharing the CPU load. RC4 tends To acquire a far greater CPU time for its processing with a small block size while the AES tends to consume much less time. However, RC4 is operating using less CPU processing time and reducing the work load on the CPU when it encrypts large data blocks.

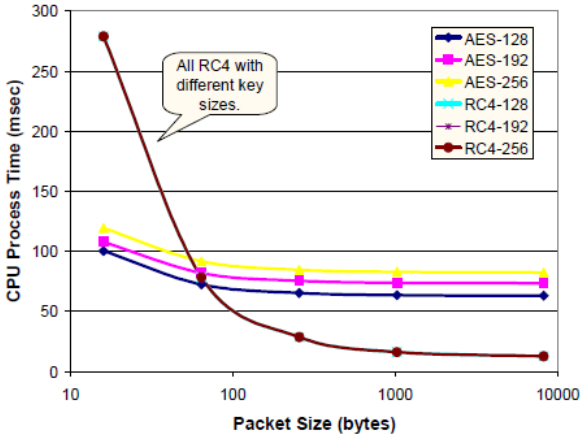

Fig 2: CPU time of RC4 and AES with varied key size

3- Integrity: for this purpose the SHA1 hashing algorithm has been implemented to maximize message integrity. In addition to integrity, speed is also considered. Therefore, this algorithm was chosen as it is considered one of the fastest. The following figure shows the performance of some hash functions.

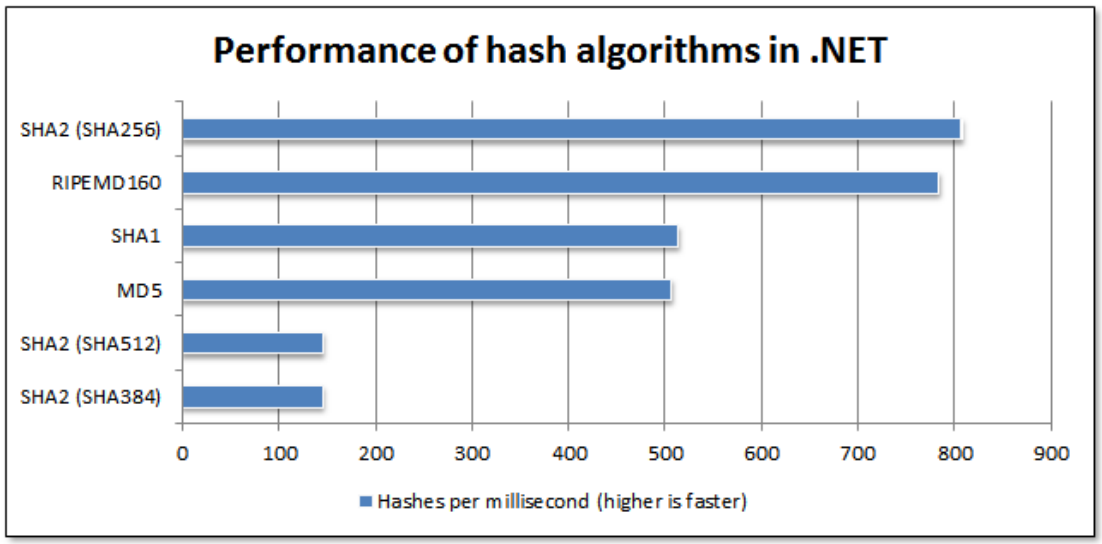

Fig 3: Hash functions performance

Table 1. Comparison between RSA and ECC

	RSA	ECC
Time of Execution		
Key Gen.	! Nondeterministic 0,2 s - 14 min 1024-bit key: 2,8 s	0,054 s - 1,4 min ECP, P+, 161-bit key: 0,09 s
Encryption	0,02 s - 6,7 s 1024-bit key, e=65537: 0,025 s	0,05 s - 2,8 min ECP, 163-bit key, 2048 byte data: 0,12 min
Decryption	0,03 s - 4,45 s 1024-bit key: 0,13 s	0,03 s - 1,55 min ECP, 163-bit key, 22 byte data: 0,05 s
Size of Data Files		
Common Params & Key Files	872 byte - 6870 byte 1024-bit key, e=65537: 1584 byte	1452 byte - 11328 byte 160-bit key, P-: 1890 byte 160-bit key, P+: 4684 byte
Encrypted Data Files	64 byte - 512 byte 1024-bit key, 22 byte: 128 byte	73 byte - 595 byte 160-bit key, 22 byte: 83 byte
Maximal. Size of Encrypted Data Files	! Strong limitation 22 byte - 470 byte 1024-bit key: 86 byte	3971 byte - 4045 byte 160-bit key: 4035 byte
Signature	64 byte - 512 byte 1024-bit key: 128 byte	30 byte - 102 byte 160-bit key: 42 byte

6. CONCLUSIONS

The problem is that the protocols tend to focus on either security or performance. Many protocols have been proposed to deal with the problems of security in both wired and wireless networks. Therefore, the proposed protocol deals with the following issues:

1- **Authentication**: establishing session keys between communicating parties is essential when using symmetric cryptographic primitives to protect confidentiality and integrity. The elliptic curve algorithm was used to provide this service with 128bit key length.

2- **Confidentiality:** two algorithms were implemented for this purpose. The first is a block cipher, the AES, with variable key length up to 1024. The second one is a stream cipher, the RC4, which is used when the high speed communication is on the board.

3- **Integrity:** for this purpose, the SHA1 has been implemented with some modifications that makes it faster.

7. REFERENCES

[1] Amjad Gawanmeh. " On the Formal Verification of Group Key Security Protocols", A Thesis, The Department of Electrical and Computer Engineering, Concordia University, 2008

[2] Benjamin Tobler. " A Structures Approach to Network Security Protocol Implementation", a Dissertation, Faculty of Science, University of Cape Town, 2005

[3] A. Freier, P. Karlton, and P. Kocher. The SSL protocol version 3.0. draft-ietf-tls-ssl-version3-00.txt, November 18 1996.

[4] T. Dierks and C. Allen. The Tls Protocol Version 1.0, 1999. RFC 2246.

[5] C. Neuman, T. Yu, S. Hartman, and K. Raeburn. The Kerberos Network Authentication Service (v5), 2005. RFC 4120.

[6] S. Kent and R. Atkinson. "Security architecture for the internet protocol", 1998. RFC 2401.

[7] IEEE P802.11i/D10.0. Medium Access Control (MAC) security enhancements, amendment 6 to IEEE Standard for local and metropolitan area networks part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications., April 2004.

[8] Wooseok Ham. " Design of Secure and Efficient E-commerce Protocols Using Cryptographic Primitives", A Thesis, School of Engineering, Information and Communications University, 2003.

[9] Gon Kim. " Formal Analysis and Automatic Code Generation of Security Protocols", A Thesis, Department of Computer Science and Engineering, Korea University, December 2004.

[10] Anupam Datta. "Security Analysis of Network Protocols: Compositional Reasoning and Complexity-Theoretic Foundation", A Dissertation, Department of computer science, Stanford university, September 2005.

[11] Sukalp Bhople. Server based DoS vulnerabilities in SSL/TLS Protocols, A Thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, August 2012.

[12] R. Needham and M. Schroeder. "Using Encryption for Authentication in Large Networks of Computers". Communications of the ACM, 21(12), December 1978.

[13] G. Lowe. "Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR. In Tools and Algorithms for the Construction and Analysis of Systems", volume 1055 of Lecture Notes in Computer Science, pages 147–166. Springer-Verlag, March 1996.

[14] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. ACM Transactions on Computer Systems, 8(1):18–36, 1990.

[15] CCITT. CCITT Draft Recommendation X.509. The Directory Authentication Framework, version 7, November 1987.

[16] O. Pereira and J. Quisquater. Some Attacks upon Authenticated Group Key Agreement Protocols. IOS Journal of Computer Security, 11(4):555–580, 2004.

[17] G. Ateniese, M. Steiner, and G. Tsudik. "New Multiparty Authentication Services and Agreement Protocols". IEEE Journal of Selected Areas in Communications, 18(4):628–639, 2000.

[18] J. Zhou. "Fixing A Security Flaw in IKE Protocols". IEEE Electronics Letters, 35(13):1072–1073, 1999.

[19] N. Ferguson and B. Schneier. "A Cryptographic Evaluation of IPSec", Technical report, Counterpane Internet Security Inc., 2000.

[20] J. Katz and J. Shin. "Modeling Insider Attacks on Group Key-Exchange Protocols", In ACM Conference on Computer and Communications Security, pages 180–189. ACM Press, 2005.

[21] O. Pereira and J. Quisquater. "On the Impossibility of Building Secure Cliques-Type Authenticated Group Key Agreement Protocols", Journal of Computer Security, 14(2):197–246, 2006.

[22] Bruce Schneier. "Applied Cryptography: Protocols, Algorithms, and Source Code in C", Second Edition.