Enhanced Authentication Protocol EAP-TTLS using encrypted ECDSA

Nazanin Bahrami [1], Mohamad Ebrahim Shiri [2], Morteza Salari-Akhgar [3]

[1] Department of Computer Engineering, Kish Azad University, Kish, Iran

[2] Department of Computer sciences Faculty of Mathematics and Computer, Amirkabir University of Technology, Hafez ave, Tehran,Iran

[3] Department of Computer Engineering, Ghorveh Branch Islamic Azad University, Ghorveh, Iran

Abstract

The growing trend of wireless networks and risks associated with them has led to the development of authentication and application of stronger cryptographic methods to maintain a more robust security. This article examined and compared two types of developed authentication protocols to provide an alternative method using strong cryptographic methods. The common EAP-TTLS method uses RSA algorithm for encryption and SHA-1 hash algorithm. In this article, the new method called alternative EAP-TTLS method uses Elliptic Curve Digital Signature Algorithm (ECDSA) and the secure hash algorithm (SHA-256) to provide stronger security and higher efficiency. This alternative method and the mechanism used in it is fully explained and compared with the common EAP-TTLS method. Results obtained indicate that the alternative algorithm provides strong security, high speed, and more applicability with the same level of memory usage compared to the common EAP-TTLS.

Keywords: *authentication, cryptography, ECDSA, SHA-256.*

1. Introduction

Increased use of local wireless networks and problems associated with these networks has led to the creation of security protocols and standards to overcome these vulnerabilities. IEEE 802.1x is one of the new standards that offers an effective framework for authentication and traffic control for a protected network. The 802.1x uses the Extensible Authentication Protocol (EAP) [1]. The Extensible Authentication Protocol is a message exchange standard that allows the server to authenticate the client using an authentication method acceptable to both sides. There are various EAP mechanisms available; one of them is EAP-TTLS, which is the mechanism in question in this study. In the common EAP-TTLS method RSA and hash SHA-1 algorithms are used, and based on the reasons explained later, in this study a new method is provided in which for EAP-TTLS, ECDSA and SHA-256 algorithms are used for higher efficiency.

1.1. Introducing EAP-TLS and EAP-TTLS protocols

The Transfer Layer Security (TLS) is a type of EAP protocol that provides the server authentication for the user, or mutual authentication of the server and user for both sides. It also negotiates encryption and key exchange between two parties. EAP-TTLS is an EAP method that provides better performance than that available in EAP-TLS.

1.1.1. TTLS negotiation

An EAP-TTLS negotiation involves two stages:

1- TLS hand-shake stage
2- TLS tunnel stage

1- TLS hand-shake stage

In the first TLS stage, TTLS has been used to authenticate server to the client and optionally, client to server. The first stage leads to activation of subsequent encryption, permitting secure processing in the second stage using TLS register layer.
Note should be taken that type and degree of security in the second stage depends on the negotiated subsequent encryption in the first stage; there is no security with a void negotiated subsequent encryption.

2- TLS tunnel stage

In the second stage, TLS register layer is used for tunneling information between the client and the server's TTLS for a number of functions. These functions may include: user authentication, verifying client integrity, key distribution and accounting data communication[2].
The main advantage of a tunneling technique is that it facilitates privatization of identifiers. With a tunnel, these techniques can protect client ID from tapping by hiding ID message to EAP response in a secure tunnel. Until authentication server has not authenticated the client in the first stage, the client can send the password through the produced secure tunnel after TLS hand shake to commence the second stage. As long as the produced secure tunnel in

the first stage hides messages sent in the second stage, the client and the server can rest assured that client authentication is as secure as EAP-TLS [3].
Today, digital certificates are the basis of most communication systems that require a high level of security between communication components, such as the EAP negotiation plan that provides secure access to wired and wireless networks.

2. EAP-TTLS mechanism and cryptographic techniques

2.1. Digital certificate

Digital certificate can be implemented in two forms:
1- **Client and server certificates**: These certificates require a PKI that every client has in his certificate. This could initially be costly because it requires a special software, training and education. In addition, creation and security of issued certificates for a client in an organization could significantly increase managerial expenditure. Also, if a system is lost or stolen, all certificates will need to be re-issued within the organization to maintain integrity.
2- **Server's certificates**: This method requires a PKI for management and distribution of certificates to servers within an organization in order to perform authentication. Servers are usually more secure than client systems, which makes management and distribution of certificates much more convenient and effective. Use of server certificates usually requires a chain or a series of certificates. Server and client certificates are often used for EAP-TLS to secure local wireless networks. Server certificates contain SSL connections to secure transactions over the internet and EAP-TTLS, and are used for securing local wireless networks.

2.1.1. An example of server certificates

Figure 1 shows certificate exchanges and verification in a transaction between client and an e-commerce on the Internet. For a client to be able to securely exchange sensitive data such as credit card details, first server ID must be verified. Then a secure tunnel is created for data transaction using Certificate Accreditation (CA).

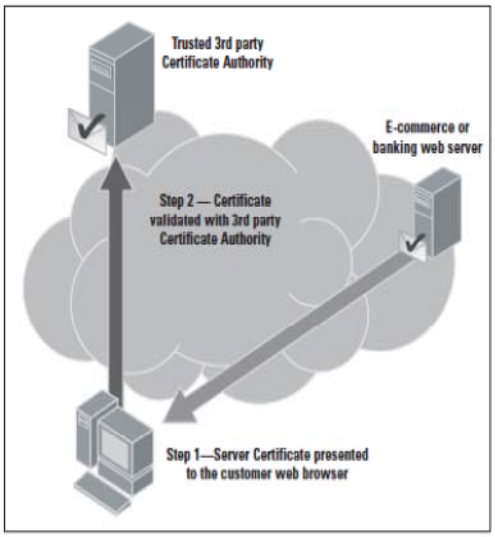

Figure 1: An example of digital certificate used in SSL for secure transactions on the internet

In EAP-TTLS a secure tunnel of transfer layer is firstly created between client and server TTLS. This tunnel supports client transactions certificates like the user name and password. When user's certificates are authenticated, a second TLS tunnel is created. Thence, key encryption information can be sent to the client.
After client has obtained key encrypted information, TLS tunnels are destroyed and all communication becomes secure using WEP. Digital certificates are very expensive for providing security and assured communication on non-secure and unsafe networks. They are extremely necessary on secure wireless network systems and transactions over the internet [4].
EAP-TTLS has been developed to overcome client certificate overheads. Using these methods, client is able to verify integrity of the certificate by reference to assured certificate Accreditation (CA). To transfer in a secure method, cryptography is an important concept that has been considered [5].

2.2. Cryptographic mechanisms in common EAP-TTLS

Protection processes used in the current EAP-TTLS depend on hash SHA-1 and RSA digital signature, and SHA-1 are used for signature purposes and assessing correctness of messages, and RSA digital signature is used for signing the exchanged messages.

2.2.1. SHA-1

Hash algorithm converts the desired size into a constant value. SHA-1 is widely used, and shortens the message to 128 bits.

Some security flaws have been identified. Theoretically, SHA-1 with less than 2^{80} hits is vulnerable to attacks. In practice, collision occurs at 2^{63} hash process, which is possible for an intruder with good resources. When one part sends a perfect protected message with SHA-1, it is possible for an invader to alter the message. Therefore, receiver receives a different message from the same hash function, and wrongly believes that message has not been altered during transfer. Because MD5 and SHA-1 have security flaws, integration of these two will not enhance security properties.

2.2.2. RSA digital signature

RSA security is strongly based on hash function and standard length. The sender signs hash message with a private key, then both message and signature are sent to another section. The second section uses received signature with the sender's public key and examines the signature. If the results are the same, signature is successfully approved. In simple words, if a text is issued from a person to others, this includes the original text plus same text but encrypted by the same person's private key. Now, if the encrypted text is deciphered by the person's public key that you know of, sameness of the resulting text and the original text confirms integrity of the sender of the text, and the person's signature is approved. Those who have no knowledge of this person's private key cannot create encrypted text and turn it back to original text using deciphering by the person's public key.
An invader can easily produce a large RSA signature key using little power. Thus, examination of key requires high power, which uses other parts of large calculation resources to examine the signature. Therefore, public key must have an acceptable power value.

3. Cryptographic mechanisms in alternative EAP-TTLS (a proposed method)

3.1. SHA-256

SHA-256 message block has 512 bits and its hash size is 256 bits. SHA-256 is one of the strong methods in SHA algorithm. SHA-256 uses 6 logic functions to work on a 32 bit word. There are a number of math defects in SHA-1 and the hash size is small. However, SHA-256 has nearly 128 collusion resistance, and its security capability has improved compared to SHA-1.

3.2. Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA is an elliptic curve analogue of Digital Signature Algorithm (DSA), and has been standardized by many world standard organizations such as: ANSI, IEEE, NIST, and ISO. In relation to ECDSA, two different signatures using two functions are produced, and used to generate points on the elliptical curve and an extractor. The first function generates points with private key, and the other function creates new points with the public key. The signature process involves: selection of a random encryption, generating one point using this encryption, and producing two signatures.
ECDSA first creates appropriate domain parameters to ensure approval of this domain. However, RSA does not produce parameters of this domain. The receiver also produces required parameters and public key through digital certificates. To produce ECDSA, area parameters, private key, and main information are created, respectively, and the mentioned hash algorithm and random number generator are used. To examine the signature, same area parameters and hash algorithms are utilized [6].

3.3. RSA versus ECDSA

Security of digital signature depends on user's private key. Extracting private key is difficult with having the public key. The deciphering problem for standard size RSA and for elliptic curve is the number of point in group work. In practice, solving elliptic curve discrete logarithm is difficult. Compared to RSA, elliptic curve provides increased storage needs, efficiency, and effective and practical use, by providing same security level with smaller system parameters and 160 bits key, compared to RSA's 1024 bits. Whilst, comparing these two methods, the implementation speed cannot be determined exactly. Elliptic curve operation is much more complex than RSA. Security of hash function is vitally important and influential for security of ECDSA, and it is equally necessary for collision resistance [7].
RSA signature is slower than ECDSA's, and RSA confirmation is faster. RSA digital signature for structure-based certificate that requires some certification production and large amount of confirmation is appropriate. Whilst, short ECDSA key is appropriate for reduced CPU wastage, reduced wireless network overhead to store broad band. RSA security depends on the factorization issue, while ECDSA is based on elliptic curve discrete logarithm.

3.4. SHA-1 versus SHA-256

According to security analysis in two digital signatures of RSA and ECDSA, two different hash functions are used. Capability and size of digests in these two methods are different, whilst both functions belong to the same family.
In the main algorithm, for security of hash function, both MD-5 and SHA-1 are concurrently used. Instead, in alternative algorithm, stronger SHA-256 and MD5 are used respectively, to provide strong hash function, for flawless hash message [8].
Strength of digital signature security is associated with the hash function used. Security of hash process must be at the same level or higher. Therefore, algorithm efficacy and

security of ECDSA must depend on SHA-256. Therefore, alternative algorithm is capable of using SHA-256 or SHA-512. As a result, alternative algorithm provides higher security using SHA-256. The 160 bit ECDSA key reduces storage requirements compared to the 1024 bit key in RSA method.

4. Simulation and results

In this section, traditional EAP-TTLS mechanism is compared with the proposed new mechanism in this article. For simulation, C#NET language, with SOKET, 3GHZ, CPU, RAM 2G programming methods have been used. In this simulation, virtual client and server have been considered in socket form, and simulation has been performed on 3 scales of: used space, execution time, and security, which are explained as follows:

4.1. Execution time

One of the EAP-TTLS problems is the peak execution time, resulting from large amount of information exchange and high calculation overhead of RSA digital signature. Using ECDSA, execution time is largely improved. Thus, this algorithm can be used in networks with weak systems. In addition, productivity is increased with this new method.

RSA and ECDSA execution times, based on number of CPU clocks, are displayed in two different modes. In the first mode, the number of clients is assumed constant and number of routers is increased in each stage, between AP and the main server.

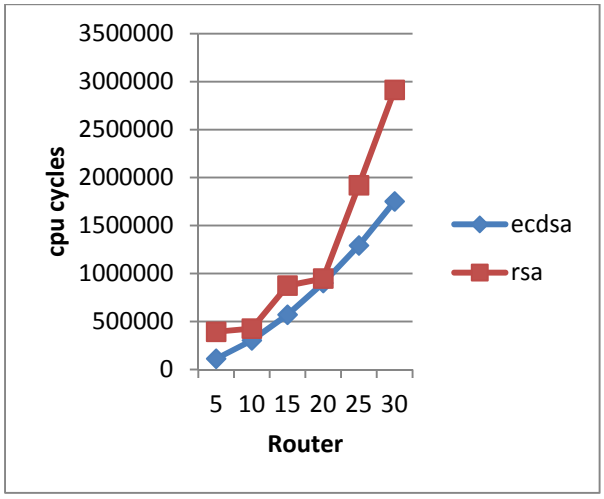

Figure 2: Comparison of RSA and ECDSA execution times with increased number of routers

Figure 2 shows that mean ECDSA execution time has improved by 44% compared to RSA.

In the second mode, the number of rotors is kept constant and number of clients is increased.

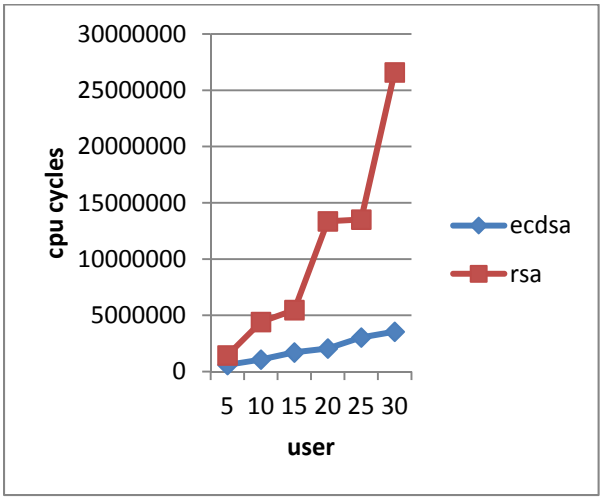

Figure 3: Comparison of RSA and ECDSA execution times with increased number of clients

Figure 3 shows that ECDSA consumption time has reduced, and this reduction is due to the low ECDSA overhead compared to RSA. Proposed method's execution time has significantly reduced.

4.2. Memory consumption

Results obtained from simulation indicate that use of two different cryptographic methods does not much affect memory consumption. Yet, use of ECDSA has caused a little improvement, compared to RSA, and this can be useful at networks peak traffic times.

To compare memory consumption between ECDSA and RSA, two different simulations were performed. In the first, number of clients was assumed constant with increasing number of rotors, and results are shown in figure 4.

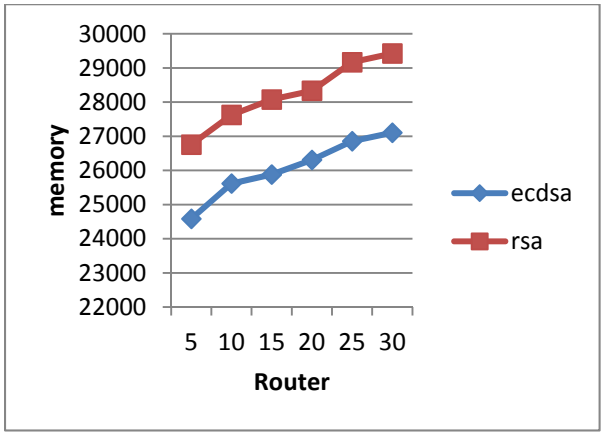

Figure 4: comparison of memory consumption with increasing routers

In this mode, memory has reduced by 8% in ECDSA compared to RSA. In the second mode, number of rotors was considered constant, and number of clients increased, as shown in figure 5.

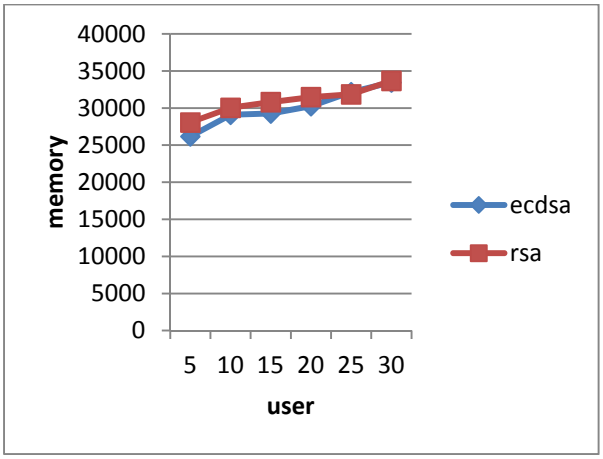

Figure 5: comparison of memory consumption with increasing clients

It can be seen from figure 5 that in this mode, and in ECDSA method, memory has reduced by 6%.

4.3. Security

The new algorithm benefited from two powerful methods for providing strong security. Exchanged information between two parties is protected by use of digital signature and hash function for providing reliability and integrity. Security of the digital signature mechanism rather depends on security of the hash function used. SHA-1 is exposed to the attackers' collision. An alternative solution is application of SHA-256 that is immune to this attack.
As ECDSA uses elliptic curve method for operations signature, attack requires knowledge of ECC implementation details to break the signature by using reverse calculation. On the other hand, to break RSA, Hacker needs a math model to perform ECDSA reverse calculations.

5. Conclusion and future works

Given the results of the simulations performed in this study, changing RSA algorithm to ECDSA and also, SHA-1 to SHA-256 in EAP-TTLS, led to improved execution time in two different modes of: constant client and constant routers. This algorithm change had less effect on memory consumption, but can be useful during peak network traffic, and causes slight improvement in memory consumption.

The new algorithm has led to improved security in EAP-TTLS due to use of SHA-256 because SHA-256, as opposed to SHA-1 is more resistant to collision attack. Therefore, use of ECDSA and SHA-256 in this alternative method could improve the method.
In future studies, this new method can be examined with use of VPN in strong cryptographic tunnel. Also, use of a memory that stores virtual users should be studied with this method.

References

[1] Rana, A.,Chillar, R.," Analysis of the Protected Extensible Authentication Protocol", International Journal of Computer Science & Management Studies, Vol.12, pp 187-191, Sept 2012.
[2] RFC 5281"Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version{EAP-TTLSV0},"ed, 2008.
[3] Han, L., "A Threat Analysis of The Extensible Authentication Protocol.", Carleton university Thesis, April, 2006.
[4] Symbol Technologies, Inc., The Enterprise Mobility Company, The use of digital certificates for authentication to a wireless LAN. February 2005.
[5] Saberi, I., Shojai, B., Salleh, M., "Enhanced Key Expansion for AEs-256 by Using Even-Odd Method", International Conference on Research and Innovation in Information Systems , Kuala Lumpur,2011.
[6] Khalique, A., Singh, K., Sood, S., " Implementation of Elliptic Curve Digital Signature Algorithm", International journal of computer Applications, Vol 2-NO.2, PP 21-27, MAY 2010.
[7] Menzes, A., Qu, M., Stinson, D., Wang, Y., "Evaluation of Security Level of Cryptography:ECDSA Signature Scheme", 2001.
[8] Shojaie, B., Saberi, I., Salleh, M., Niknafskermani, M., Alavi, S M., "Improving EAP-TLS Performance Using Cryptographic Methods", IEEE, 2012 International Conference on Computer & Information Science.

First Author Nazanin Bahrami received her master`s Degree from the department of computer Engineering at the Islamic Azad university Kish International Branch in 2013. She is a member of Iranian society of computer. Her research interests about the network security and intrusion detection systems.

Second Author Mohammad Ebrahim Shiri is an assistant professor in the department of computer sciences at Amirkabir University of Technology of Tehran, Iran. He received his Ph.D. from the department of computer sciences at the University of Montreal, Canada in 1999. His current research interests include artificial intelligence, multi-agent systems, intelligent tutoring systems and distributed systems.

Third Author Morteza Salari Akhgar has a Master`s Degree in software engineering. He is a researcher and author of several books on computer security.