Research Journal of Applied Sciences, Engineering and Technology 12(5): 550-555, 2016
DOI:10.19026/rjaset.12.2578
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 21, 2015 Accepted: November 11, 2015 Published: March 05, 2016

Research Article

A Lightweight Digital Signature Cryptographic Protocol for Authentication and Integrity Based on Location

[1]Gina Gallegos-Garcia, [1]Raul A. Flores-Espinoza, [2]Horacio Tapia-Recillas, [1]Alejandro Valdez-Aguilar and [3]Gualberto Aguilar-Torres
[1]Instituto Politecnico Nacional, ESIME Culhuacan, Av. Sta Ana. 1000. Sn. Fco. Culhuacan, 04430,
[2]UAM Iztapalapa. Sn. Rafael Atlixco 186, Vicentina, Iztapalapa, 09340,
[3]Comision Nacional de Seguridad. Secretaria de Gobernacion. Av. Constituyentes #947 Col. Belén de las Flores, Del. Álvaro Obregón, Distrito Federal CP. 01110, Mexico

Abstract: The shipping of digital documents, from a sender to a receiver, must keep certain properties, including the authenticity of the source and message integrity. These properties can be achieved through cryptographic protocols, which define the way in which a cryptographic primitive should be used during a communication process. As an additional function to the mentioned properties, GeoLock is a function that is used to add the coordinates of a receiver to the digital signature primitive or to other cryptographic primitives, which helps to prove that such entity is located in the right place and is able to verify a generated signature avoiding antispoof receivers. However, the use of public key cryptographic primitives with the GeoLock function on mobile devices, involves considering the performance of all cryptographic operations together. Considering the aforementioned, in this study a lightweight cryptographic protocol based on the digital signature primitive and GeoLock is presented in order to meet authentication and integrity based on location. The obtained results show that the proposed protocol is more efficient if compared to related works, as less cryptographic operations are required.

Keywords: Cryptographic protocol, digital signature, geolock, mobile device

INTRODUCTION

Cryptographic protocols are communication distributed algorithms that define how a set of primitives should be used by two or more entities that exchange messages to achieve the traditional information security goals: confidentiality, integrity and authentication (Menezes *et al*., 1986). Cryptographic protocols must be carefully designed due that they can contain several types of flaws and vulnerabilities that could be exploited by intruders.

Nowadays, with the rapid growth of the use of mobile devices, different cryptographic protocols have been proposed in order to provide needed security to mobile devices, such as: the use of digital signatures as cryptographic primitives to meet authentication and integrity of interchanged messages. Moreover, location services technology is being combined with digital signatures to prove that the receiver is in the right place to verify the message signature (Lei *et al*., 2004; Jarusombat and Kittitornkun, 2006).

Proposals made until now have used the RSA algorithm as a digital signature primitive on mobile devices. This brings concerns about the performance of cryptographic operations in mobile devices, due to the low-computation capability and limited battery life of these devices. Considering the aforementioned, in this study a lightweight cryptographic protocol is proposed to achieve authentication and integrity based on location. The proposed protocol is based on the digital signature primitive and GeoLock function. Moreover, due to its functionality, which is based on additive groups, the proposed protocol is more efficient if compared to related works, as less cryptographic operations are needed.

MATERIALS AND METHODS

Our proposed protocol considers three main research variables: Efficiency, digital signature and GeoLock function. All of them are defined in following subsections.

Corresponding Author: Gina Gallegos-Garcia, Instituto Politecnico Nacional, ESIME Culhuacan, Av. Sta Ana. 1000. Sn. Fco. Culhuacan, 04430, Mexico

Efficiency: Efficiency can be seen as the amount of resources used by a cryptographic protocol. In this sense, a protocol that is being analyzed from the efficiency point of view should determine its resources usage, where a resource can be: time, memory space or cryptographic number of operations, among others.

Digital signatures: There are three different types of cryptographic primitives:

- Unkeyed primitives
- Symmetric-key primitives
- Asymmetric-key primitives

In symmetric-key primitives there exists a single key associated to two different operations, encryption and decryption (E, D). In asymmetric-key primitives each key pair is associated with encryption and decryption operations (E, D). One of the keys is denoted by k_{public} and is known as public key due that it is available to the public. The other key $K_{private}$, called private key, is kept in secret. It is worth mentioning that this last type of primitive has the characteristic that is based on the knowledge of the public key and is computationally difficult to calculate its private counterpart (Menezes *et al*., 1986).

A digital signature, which for example can be a symmetric or an asymmetric-key primitive, is a string of data that associates, in a digital way, a message with the entity that is issuing it, also called the sender or signer. The fact of associating the identity of a sender with the message, helps to ensure, to a third entity, that the signer cannot deny the action of have signed the message. In addition to that, it allows knowing if the message has suffered alterations after having been signed. Moreover, it helps to assure the receiver that the sender is who he/she claims to be (Menezes *et al*., 1986).

The entities involved in the process of cryptographic signature, commonly called digital signature are: the sender also called the signer and the receiver also called the verifier. The sender is responsible of generating the digital signature. The receiver receives the signed document and verifies its corresponding digital signature (Menezes *et al*., 1986).

GeoLock: The GeoLock mapping is a function computed on the basis of the intended receiver position, velocity and time, referred to as the PVT block, which converts all those parameters into a unique value as a lock. The PVT block defines where the receipt must be in terms of its position, velocity and time for a cryptographic operation to be successful. In other words, such unique lock value validates that the receivers satisfy certain restriction, for example the encryption, decryption, signature generation or verification region at a certain time interval, without having preinstalled mapping tables (Scott and Denning, 2003).

CONSIDERATIONS OF THE PROPOSED MOBILE CRYPTOGRAPHIC PROTOCOL

The efficiency: The design of our cryptographic protocol considers as a goal to be efficient when executed in mobile devices. It is important to emphasize that the main improvement in our protocol consists in providing an efficient performance by developing less cryptographic operations.

In order to develop a digital signature in mobile devices, the proposed protocol is based on the functionality of additive groups. As a consequence, it uses smaller keys, in comparison to other algorithms commonly used, such as RSA (Rivest *et al*., 1978). Moreover, the proposed protocol bases its security on the Elliptic Curve Discrete Logarithm Problem (Hankerson *et al*., 2003).

The certification authority: As in Jarusombat and Kittitornkun (2006), the design of our proposed protocol considers a Certificate Authority (CA), which is private and was designed in order to wait for a personalized Digital Certificate request from the sender. To do this, such sender sends a certificate request to the CA. A certificate request is built with the CA's public key, the number of valid signatures (that the certificate will support), the name of the signature server (that will help mobile devices to generate digital signatures) and the sender data identification. In this way, the CA will be able to validate keys generated by the sender and will ensure that those ones are correct. Then, the CA will generate a certificate that will be sent to the sender, which in turn will use it in order to be a valid sender in front of receivers (Barker *et al*., 2012; Lee, 2010; Zhang *et al*., 2008).

The use of GeoLock function: In order to add the coordinates of the receiver into the digital signatures generation, the receiver and the signature server applies the GeoLock mapping function. After GeoLock manipulates the coordinates of the receiver, a hash value is obtained from the result, in order to preserve the integrity property of the location coordinates. Thus, the signature is generated by using the location coordinates, manipulated previously. When signature verification is performed, the receiver can verify if the location coordinates have been changed. If the integrity of the location coordinates is maintained, then it will be possible to ensure that receiver will be in the indicated place at the time of receiving the signature; otherwise, the digital signature cannot be verified. Moreover, these computations are completed by the receiver and the signature server in order to xor the coordinates with the rest of the data that are involved in the digital signature generation and its verification.

The proposed mobile cryptographic protocol: The proposed efficient mobile cryptographic protocol is divided into three stages: Key generation, digital

signature generation and digital signature verification, which are detailed bellow.

Key generation: It is carried out twice, the first one is performed by the sender to obtain a key pair denoted by (d_{snd}, Q_{snd}). The second one is carried out by the server to obtain a key pair denoted by (d_{srv}, Q_{srv}):

- Given $q, a, b, P \in E(F_q)$ and n, where q is the field order, $a, b \in F_q$ that define the equation of the elliptic curve E over F_q, the sender selects $d_{snd} \in [1, n-1]$ and computes (Q_{snd}) as his public key, according to Eq. (1). Such key is sent to the Certificate Authority (CA) to obtain a digital certificate:

$$Q_{snd} = d_{snd} \quad (1)$$

- The signature server executes the same process executed by the sender and selects $d_{srv} \in [1, n-1]$ and computes (Q_{srv}) in a similar way as is given in Eq. (1). As a consequence, a second key pair is obtained. After that, the server sends to the sender the public key (Q_{srv}), the sender then uses the signature server's public key to generate the values needed to generate a digital signature. The private key that was generated by the signature server, denoted by (d_{srv}), is used to help the sender to generate the digital signature, even though the sender does not know the server's private key.

Signature generation: It is performed by the sender with the help of the signature server, following the next five steps:

- The sender selects two random numbers (r, w) and computes k, based on Eq. (2), where $(*)$ represents the operation of multiplication and h denotes a cryptographic hash function whose outputs have a bitlength no more than n. Then, k is used to compute g, based on Eq. (3), where x_1 is converted to an integer and $g = x_1 mod n \rightarrow \hat{x}_1$:

$$k = r * w * h(x) mod n \quad (2)$$

$$kQ_{srv} = (x_1, y_1) \quad (3)$$

- The signature server computes $Z \in [1, n-1]$. It is sent to the sender. In addition to that, the signature server receives the coordinates mapped with the GeoLock function at which the receiver is currently located.
- The sender selects a random number (r') to compute b, based on Eq. (4), with $(Z_n^*, *)$ and uses b to multiply P_{srv}. Also is computed η, according to Eq. (5), where x_2 is converted to an integer and $x_2 mod n \rightarrow \hat{x}_2$. After that, the coordinate x_2 of η is sent to the server:

$$b = r * r' * Z * g mod n \quad (4)$$

$$\eta = bP_{srv} = (x_2, y_2) \quad (5)$$

- x_2 is used to calculate j and after calculating j, t is computed, based on Eq. (6) and (7), respectively. To do this, the signature server uses its private key to combine it with the hash value of the xored sender's certificate and the mapping function with the coordinates of the receiver. After that, t is sent to the sender:

$$j = x_2 mod n \rightarrow \hat{x}_2 \quad (6)$$

$$t = h(x) + h(Cert_{snd} \oplus GeoLock_{rcv}) + d_{srv} j \quad (7)$$

- With the data received from the signature server and using Eq. (8), the sender then calculates the last parameter of the digital signature, denoted by S. Finally, the signature server receives S from the sender, where $S \in \mathbb{Z}_n^*$:

$$S = b^{-1}(t) mod n \quad (8)$$

Signature verification: It is carried out twice and follows four steps. The first time it is performed by the server, before sending the signature to the receiver. The second time, it is carried out by the receiver:

- Based on Eq. (9), the sender calculates f:

$$f = S^{-1} mod n \quad (9)$$

- With f, the server calculates u_1, u_2 and u_3, based on Eq. (10), (11) and (12), with x as the message:

$$u_1 = h(x) f mod n \quad (10)$$

$$u_1 = h(x) f mod n \quad (11)$$

$$u_3 = j f mod n \quad (12)$$

- Finally, to verify the digital signature, the server calculates X based on Eq. (13), where $y = x_3 mod n \rightarrow \hat{x}_3$ and x_3 is converted to an integer. Then, check if $y == j$. If y is equal to j, then the digital signature is true, so the server sends the signature along with others parameters to the receiver to verify the signature one more time:

$$X = u_1 P_{srv} + u_2 P_{srv} + u_3 Q_{srv} = (x_3, y_3) \quad (13)$$

- Once the receiver receives the digital signature, his/her mobile device carry out the second verification with the same steps previously detailed and that the signing server should follow. This time, the receiver calculates the coordinates where he/she is located at the time at which he/she

receives the signature. In this way, we ensure that the receiver has not been moved from the place where he/she should be. In other words, if the receiver is located at different coordinates from those one at which he/she should be, then the digital signature cannot be verified as true. The unique change performed in the process to verify the digital signature is in the variable u_2, based on Eq. (14):

$$u_2 = h(Cert_{snd} \oplus GeoLock_{rcv2}) f mod n \quad (14)$$

Correctness proof of the proposed lightweight cryptographic protocol:
From Eq. (8), we have that:

$$S = b^{-1}(t) mod n$$
$$b \equiv S^{-1}$$
$$\begin{pmatrix} h(x) + h(Cert_{snd} \oplus GeoLock_{rcv}) \\ +d_{srv} j \end{pmatrix} mod n$$
$$\equiv S^{-1}h(x) + S^{-1}h(Cert_{snd} \oplus GeoLock_{rcv})$$
$$+S^{-1}d_{srv} j mod n$$

From Eq. (9), (10), (11) and (12), we have that:

$$b \equiv fh(x) + fh(Cert_{snd} \oplus GeoLock_{rcv})$$
$$+f d_{srv} j mod n$$
$$b \equiv u_1 + u_2 + u_3 d_{srv} mod n$$
$$X \equiv u_1 P + u_2 P + u_3 d_{srv} P$$
$$\equiv u_1 P + u_2 P + u_3 Q_{srv}$$
$$\equiv bP$$
$$\therefore y = j$$

RESULTS AND DISCUSSION

It is important to say that we analyze efficiency is seen from the point of view of the cryptographic operations needed to perform the proposed protocol. Table 1 and 2 show the number of modular multiplications, selection of random numbers, exponentiations and multiplicative inverse operations developed in the proposed lightweight cryptographic protocol.

As we can see, the difference between the number of inverse multiplicative operations that have to be computed has a relation 1:3.75. However, it can be decreased with a reduction in the number of exponentiations developed, which is of 100%. Moreover, multiplications and selection of random numbers are minimized 78% and 71%, respectively. Considering the above reductions we can say that a lightweight digital signature cryptographic protocol for authentication and integrity based on location has been proposed.

If we consider the number of cryptographic operations carried out in order to execute the proposed protocol, then we can confirm that fewer operations are needed in comparison to those required to execute the approaches presented in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006), as can be seen in Table 1 and 2. However, it is important to notice that the number of modular multiplications required in the proposed approach is almost twice the number of those required in the protocols proposed in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006). This is because during signature verification, the proposed protocol performs different operations, that require more processing on the mobile devices, which do not happen in the approaches (Lei *et al*., 2004; Jarusombat and Kittitornkun, 2006). This is, in the verification process, the protocols proposed in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006) need only one modular operation, while our proposed protocol needs six different modular operations. The use of additive groups is the reason why more modular operations are required.

Another important point to highlight is the absence of exponentiation operations in the proposed protocol. Hence, the computation of the modular multiplicative inverses in the proposed protocol requires more computations than the computations required in the protocols prosed in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006). However, if we make a comparison between exponentiation operations and modular multiplicative inverses, our statements are the follows: In modular multiplicative inverse, there are two numbers that have the same length. The fact of finding the inverse of one of them does not require much computational resource in mobile devices. This, if is compared with exponentiation operation. In this last one it is necessary to raise a number to an exponent, resulting in a different number larger than the original

Table 1: Exponentiations vs modular inverse multiplicative operations in the proposed protocol

Operation	Lei	Santi	Proposed protocol
Exponentiation	13	13	0
Inverse multiplicative	4	4	15

Table 2: Exponentiations vs modular inverse multiplicative operations in the proposed protocol

Operation	Lei	Santi	Propose Protocol
Multiplication	13	14	11
Random numbers	6	7	5

Table 3: Performance of proposed protocol has a reduction of 43% and 52%

Performance	Lei (ms)	Santi (ms)	Proposd Protocol
Time execution	11.87	14.37	6.2 ms

one. Therefore, considering that the size of the numbers we use is smaller, we conclude that the proposed protocol is more efficient than the protocols proposed in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006). In addition, considering that point doubling in additive groups is a little bit similar to exponentiation in multiplicative groups, the efficiency of the proposed protocol is better than that achieved by the protocols proposed in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006).

The aforementioned can be seen, from the implantation point of view, when the receiver verifies the digital signature, because the time used is larger compared to that used in the protocols proposed in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006). This is because the signature verification makes scalar multiplications and point adding operations in finite fields, which are heavy processing operations, in computer terms, when performed in mobile devices. However, taking into account the total time spent in completing the whole process: key generation, signature generation and signature verification, the time we get in our protocol is much lower than the one used in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006) protocols. In those approaches during key generation it is necessary to find a relative prime number of such value but of 1024 bits in size, known as Φ(n). As a consequence, their protocols perform more operations in comparison to the number of operations carried out in our proposed protocol. In addition, despite of Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006) is being using a very small public exponent, they consider p, q, n, d as variables of around 1024 bits. Moreover, as a study case we codify, in an object-oriented progamming language, proposed protocols in Lei *et al*. (2004) and Jarusombat and Kittitornkun (2006). It was made by using the same parameters as they say. After that, we also codify our proposed cryptographic protocol and compare it. In that codification, sender and receiver were implemented in a Samsung Galaxy Young, that is a device with a 832 MHz processor, 290 MB of RAM and Operating System Android v2.3 called Gingerbread. Signature server was implemented in a laptop with Windows 7 Operating System and Certificate Authority was created on a personal computer with Ubuntu 12.04 LTS Operating System. It is important to mention that in our study case the scenario was deployed in a wireless local network. Table 3 shows that proposed protocol has a reduced performance of 43 and 52%. Both of them compared against (Lei *et al*., 2004; Jarusombat and Kittitornkun, 2006) protocols.

CONCLUSION

Nowadays the use of mobile devices is part of poeple's daily life, which gives them the opportunity to send and receive a lot of messages. The security of such messages and its owner are important points to be focused on.

The uses of digital signature as the main construction block of a cryptographic protocol, to assure, on the one hand, the source and receiver authenticity and on the other hand, the message integrity, should be performed from anywhere in which the signer is located. This process should not require the use of a personal computer, therefore the digital signature on mobile devices is fundamental and practical.

In this study, an efficient mobile cryptographic protocol based on digital signature primitive and GeoLock was presented. The proposed protocol works over additive groups; as a consequence the complete process is developed with shorter key-lengths than related works.

The use of additive groups in the proposed mobile cryptographic protocol let us the advantage of performing less cryptographic operations, meaning that the cryptographic protocol is executed in an efficient way, considering the cryptographic operations as the resource to be reflected on.

Finally, it is important to say that even by using the GeoLock function, to add an extra layer of security focused on authentication and integrity based on location, the efficiency of the proposed protocol is preserved.

ACKNOWLEDGMENT

The authors thank the Instituto Politecnico Nacional and the Consejo Nacional de Ciencia y Tecnologia. The research for this study was financially supported by Project Grant No. SIP-2014-RE/123, CONACyT 216533.

REFERENCES

Barker, E., W. Barker, W. Burr, W. Polk and M. Smid, 2012. Recommendation for Key Management-Part 1: General (Revision 3). NIST Special Publication 800-57. Computer Security Division (Information Technology Laboratory), National Institute of Standards and Technology, Gaithersburg MD 20899-8930.

Hankerson, D., A.J. Menezes and S. Vanstone, 2003. Guide to Elliptic Curve Cryptography. Springer-Verlag New York, Inc., Secaucus, NJ, USA, pp: 1-147.

Jarusombat, S. and S. Kittitornkun, 2006. Digital signature on mobile devices based on location. Proceeding of the International Symposium on Communications and Information Technologies, pp: 866-870.

Lee, B., 2010. Unified public key infrastructure supporting both certificated-based and ID-based cryptography. Proceeding of the International Conference on Availability, Reliability and Security. Dept. of Information Security, Joungbu University, pp: 54-61.

Lei, Y., D. Chen and Z. Jiang, 2004. Generating digital signature on mobile devices. Proceeding of the 18th International Conference on Advanced Information Networking and Applications, pp: 532-535.

Menezes, A.J., S.A. Vanstone and P.C. van Oorschot, 1986. Handbook of Applied Cryptography. CRC Press Inc., Boca Raton, FL, USA, pp: 4, 5, 11, 12, 425, 426.

Rivest, R.L., A. Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21(2): 120-126. http://dl.acm.org/citation.cfm?id=359342.

Scott, L. and D.E. Denning, 2003. Geo-encryption: Using GPS to enhance data security. GPS World, April 2003, pp: 40-49. http://calhoun.nps.edu/handle/10945/37168.

Zhang, Z., W. Susilo and R. Raad, 2008. Mobile ad-hoc network key management with certificateless cryptography. Proceeding of the 2nd International Conference on Signal Processing and Communication Systems (ICSPCS, 2008), pp: 1-10.