A Digital Signature Scheme using Diffie-Hellman Key Exchange

Muhammad Fareed Uddin and Kashif Siddiqui
SZABIST
Karachi, Pakistan

Abstract: *The new age digital communication has drastically changed the way we communicate and conduct business with each other. This entire electronic mode of doing communication and business follows one basic principal: Exchange of electronic message between the sender and the receiver. As technology is growing at an exponential pace, so are the threats of electronic frauds and forgery. Therefore it becomes crucial for the sender/receiver to verify whether the message they are receiving is genuine, authentic, and is actually from the source which it claims to be. To authenticate digital messages, computer scientists have come up with a brilliant idea: Digital Signatures. There are plethora numbers of techniques to generate and verify digital signatures, but they all follow one basic theme: Message fingerprinting. This paper illustrates the basic operations behind RSA digital signature and compares it with a relatively new technique of Diffie Hellman digital signature scheme.*

Keywords: *Diffie Hellman, Key Exchange, RSA, Security, Public Key Cryptography, Digital Signatures, Message Authentication*

1. INTRODUCTION

Over the past few decades, digital communication has drastically changed the way we communicate and conduct business with each other. For instance, in our day-to-day lives, email has almost completely replaced handwritten mail. Traditional banking transactions, such as payment and remittance, have been completely digitized. Credit/Debit cards are preferred over carrying cash or checkbooks and is also considered secure.

This entire electronic mode of doing communication and business follows one basic principal: Exchange of electronic message between the sender and the receiver. As technology is growing at an exponential pace, so are the threats of electronic frauds and forgery. Therefore it becomes crucial for the sender/receiver to verify whether the message they are receiving is genuine, authentic, and is actually from the source which it claims to be.

Traditionally, such forms of authentication were done by using handwritten signatures, paper seals, letterheads, etc. However, such traditional methodologies for message authentication cannot be applied to digital messages as handwritten signatures and paper seals can easily be forged using graphics tools leaving no mark of foul play.

To authenticate digital messages, computer scientists have come up with a brilliant idea: Digital Signatures. There are plethora numbers of techniques to generate and verify digital signatures, but they all follow one basic theme: Message fingerprinting. A fingerprint of the message, which needs to be transmitted, is generated, and then encrypted using some form of key (symmetric or asymmetric) which only the sender or the receiver knows. This encrypted fingerprint of the message is sent along with the message to the receiver. On receiving the message, the receiver decrypts the encrypted fingerprint and generates its own fingerprint of the message. If both the fingerprints match, the message is authentic otherwise, it has been tempered.

2. SYMMETRIC CRYPTOGRAPHY

Symmetric key or single key cryptography refers the form of cryptography in which a single key is used to encrypt and decrypt messages. This secret key must be known to the sender and the receiver. For a long time, symmetric key cryptography was the only form of cryptography and is still used in many applications.

The strength of symmetric key cryptographic algorithm directly depends on following factors:

1. The complexity of the cryptographic algorithm.
2. The key size.

The secret key is created independently of the data which needs to be encrypted/decrypted. This is to make sure that a cryptanalyst, who happens to intercept on or more cipher text, is unable to derive the key from the cipher text [1]. The length of the key must be such that a brute force attack, an attack which tries all the possible combinations of the key, fails in following scenarios:

1. The useful life of the message.
2. The cost of decryption exceeds the benefits gained from decrypting the message.

The advantages of symmetric key cryptography are:

1. Smaller key size (as compared to asymmetric keys).
2. Ubiquity of applications and devices that implement symmetric key cryptographic algorithms.

The disadvantages of symmetric key cryptography are:

1. The secret key must be shared across all communicating parties.
2. Provides no security against non-repudiation.
3. If the secret key is lost, all encrypted communication is compromised.

The popular symmetric key cryptographic algorithms in used today are 3DES and AES.

3. ASYMMETRIC CRYPTOGRAPHY

As discussed in the previous section, the disadvantage of symmetric key cryptography is the sharing of the secret key among the communicating parties. The rate of growth of internet surpassed everyone's expectation. These days, people not only use internet via their desktop terminals but also connect to it via tablet PCs, smart phones, handheld devices, mobile phones etc. Moreover, these devices are also used to make business transactions. For instance, it's not uncommon that people pay their utility bills via their mobile phones or people do online shopping. As discussed, these business transactions need to be conducted over a secure channel to avoid any eavesdropping.

With the internet boom, it became more and more difficult to exchange secret key before initiating a secure communication. The solution came in the form of public key or asymmetric key cryptography. In this form of cryptography, two separate keys are used for encryption and decryption namely the public key and the private key.

The encryption/decryption operations of asymmetric key cryptography can be explained in the following steps:

1. User A generates a pair of related keys: A public key, and a private key.
2. The private key is held securely by the user, and never leaves his/her possession.
3. The public key is published and is known to everyone.
4. If User B wishes to send a secure message to User A, it encrypts the message using User A's public key (since it is known).
5. On receiving the message, User A decrypts the message using his/her private key.
6. NOTE: Message encrypted with User A's public key can ONLY be decrypted using his/her corresponding private key.

   The most widely used asymmetric key cryptographic algorithm in use today is the RSA.

4. KEY EXCHANGE

There are numerous applications that still rely on traditional symmetric key cryptography as main provider of security. These applications cannot be modified to use asymmetric key cryptography due to one or more of the following reasons:

1. The cost of upgrading is too high.
2. The symmetric key algorithm is hard-wired into the device.
3. The processing power of application or device is limited, therefore, it can only store and process symmetric key cryptographic keys.

As discussed in the previous sections, the biggest challenge and drawback of symmetric key cryptography is the secure key exchange between the communicating parties.

4.1 DIFFIE HELLMAN KEY EXCHANGE

To overcome the problem, two computer scientists: Whitfield Diffie and Martin Hellman, published a scheme, called Diffie Hellman Key Agreement in 1976. The major goal of Diffie Hellman key exchange is that two parties should be able to share a secret key without actually transferring the key. Each should be able to calculate exactly the same secret key.

Following mathematical operations are performed to exchange the secret information:

1. User A and User B agrees upon a prime number q and a primitive root α.
2. User A chooses a secret private key, PR_a, such that $PR_a < q$.
3. User A then calculates the respective public key, PU_a:

$$PU_a = \alpha^{PRa} mod\ q \quad (1)$$

4. User A then sends its public key, PU_a, to User B.
5. User B also chooses a secret private key, PR_b, such that $PR_b < q$.
6. User B then calculates the respective public key, PU_b, such that:

$$PU_b = \alpha^{PRb} mod\ q \quad (2)$$

7. User B must make sure that $PU_b \neq PU_a$
8. User B then shares his/her public key with User A.
9. Now both the parties can calculate their common secret key, K:

$$K = {PU_b}^{PR_a} mod\ q \quad (3)$$

$$K = {PU_a}^{PR_b} mod\ q \quad (4)$$

In this way, common secret key can be shared among the two communicating parties without exchanging the actual key. Each party can calculate the secret key and will get exactly the same result. Do note that the calculation of the secret key for any party depends on the public key of the other party.

5. DIGITAL SIGNATURES

The smartest solution to the problem of conventional message authentication was digital signatures. Digital signatures must possess the following attributes:

1. Verification of date and time and author.
2. Authentication of content.
3. Can be verified by a neutral third party in case of dispute.

Considering these properties, we can devise the subsequent requirements for a digital signature:

1. Signature must generate a pattern of bits corresponding to the message i.e. by using some hashing mechanism to generate finger-prints.
2. Some unique information pertaining to the sender and the receiver.
3. Easy to re-produce for verification.
4. Practically impossible to forge.
5. Digital copy can be stored.

Let us first consider a traditional digital signature scheme using the RSA algorithm:

5.1 RSA DIGITAL SIGNATURE

RSA digital signature scheme uses the RSA key of the sender and a hashing algorithm.

The digital signature generation works as follows:

1. Alice wants to send a message M to Bob, who could later verify it.
2. Alice generates her set of private and public keys: PU_a *and* PR_a.
3. Alice generates a hash H of the message M using any hashing cryptographic algorithm, such as SHA-1 or SHA-168. Hash acts as a fingerprint of the message M. If even a single bit of message M is modified, the hash H is invalidated:

$$H = Hash(M) \quad (5)$$

4. Alice then encrypts the hash H using her private key to create a digital signature S:

$$S = E_{PRa}(H) \quad (6)$$

5. Alice then appends the signature S to the message M, and sends it to Bob.

The digital signature verification on Bob's end works as follows:

1. Bob receives the message M along with the digital signature S.
2. Bob uses Alice's public key, PUa to decrypt the signature and get the sent hash value V:

$$V = D_{PUa}(S) \quad (7)$$

3. Bob computes the hash of message M and computes his calculated hash with the hash received in the digital signature. If the two hashes match, the signature is valid, otherwise not.

The RSA digital signature has one drawback, however, its key size.

5.2 DIFFIE HELLMAN DIGITAL SIGNATURE

Here I represent a new digital signature scheme based on the Diffie Hellman Key Exchange mechanism. The technique works in two steps: First the two communicating parties must exchange a symmetric cryptographic key, such that of 3DES or AES. Next, the digital signature is generated/verification is done using this secret symmetric key.

The key exchange mechanism works as follows:

1. Alice and Bob create their own set of private and public keys using Diffie Hellman Key Exchange mechanism stated earlier. Keys of Alice: {PUa, PRa}, Keys of Bob {PUb, PRb}
2. They also able to compute a common 3DES key, K.

Signature generation works as follows:

1. Alice wants to send an authentic message M to Bob.
2. Alice computes hash H of the message M, and generates a digital signature S using the common secret key K:

$$S = E_k(H) \quad (8)$$

3. Alice then appends the hash H and the signature S to the message and sends it to Bob.

The digital signature verification process works as follows:

1. On receiving the message, Bob computes the hash of the message M.
2. It then obtains the common secret K from Alice's public key. To avoid secret key generation at real time, the secret keys must be stored as some secured location for each public key. Bob then decrypts the signature using the common secret key K, to obtain the hash sent by Alice.
3. Bob compares the two hashes. If both are same, then the signature is valid, otherwise, signature is invalid.

5.3 DH SIGNATURES WITH HASH

To make the mechanism further secure, both the parties can create a value appending their public and private keys and generate hash of this value. This hash containing the public and private value will act as a unique cryptogram for the communicating parties:

Alternatively, during the digital signature generation and verification process, this hash can be embedded with the hash of the message.

For example, when user A wants to send a secret message to user B, it follows the same procedure as describe earlier, but also appends the cryptogram of User B's secret key before encrypting it with the common key:

On the other side, when receiver receives the message, it decrypts signature using the common secret key to verify whether it contains it's cryptogram or not. In this way, receiver is confident that the message indeed arrived from the intended sender.

6. TEST SETUP

As mentioned in the objective, the purpose of this research is to compare the performance of digital signatures generated using RSA and the Diffie Hellman mechanism.

For RSA, a 1024-bit key is used. As stated in the earlier sections, the SHA-1 hashing algorithm generates a hash of 160-bits. Since we are using two hashes: One for the actual message and one for the receiver's private and public key; the actual message size will be 320-bits. As RSA requires the message to be of the same size as the key, random bits have been appended with the message to make it 1024-bit long.

For 3Des, a 192-bit key is used. Since the message size is 320-bits, which matches the 3Des' block cipher size, no extra padding is required.

7. TEST RESULTS

RSA Key Size (bit)	Msg Size (bit)	Enc (ms)	Dec (ms)
1024	1024	48.8±2.9	5045.9±5.2
192	320	10.5±1.0	10.5±1.0

8. CONCLUSION

It's evident from the test results that 3DES offers better performance, both in terms of message size and encryption/decryption speed, than the RSA. However, digital signature generation using a symmetric key is relatively a new technique is RSA is proven in this area and is used in many applications. However, the Diffie Hellman Key exchange also involves overhead of key exchange and key hash calculation and exchange before a secure communication can take place.

9. REFERENCES

[1] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 22(6):644–654, 1976.

[2] Sound computational interpretation of formal encryption with composed keys by P. Laud and R. Corin. In proceedings of 6th International Conference on Information Security and Cryptology ICISC'03, volume 2971 of LNCS, pages 55-66, 2003.

[3] Proceedings of Crypto by D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In Matt Franklin, editor, LNCS, 2004.

[4] An efficient system for non-transferable anonymous credentials with optional anonymity revocation by J. Camelish and A. Lysyanskaya. In Proceedings of Eurocrypt'01, 2001.

[5] Unforgeable encryption and adaptively secure modes of operation by J. Katz and M. Yung. In Fast Software Encryption, FSE'00, volume 1978 of LNCS, pages 284-299, 2000.

[6] A method for making password-based key exchange resilient to server compromise by Craig Gentry, Philip MacKenzie, and Zulfikar Ramzan. In Cynthia Dwork, editor, CRYPTO 2006, LNCS, pages 142–159. Springer-Verlag, Berlin, Germany, August 2006.

[7] W. Stallings, "networks security and cryptography, fourth edition,2001

[8] Yacine Rebahiprdi Jaen Pallares,Gergely Kovacs,Dorgham Sisalem "Performance Analysis of Identity management in the session Initiation Protocol" IEEE Journal. 2002.

[9] Elliptic Curve Cryptography, an Implementation Tutorial by Anoop MS, Tata Exlsi Ltd, India

[10] G.J. Simmons - The Science of Information Integrity Contemporary Cryptology, 1992.

[11] Werner Schindler. A timing attack against RSA with the Chinese Remainder Theorem. In CHES 2000, pages 110{125. Springer, 2000.

[12] Vivek Kapoor, Vivek Sonny Abraham and Singh, Elliptic Curve Cryptography Issue 20.

[13] Ganesh Ananthanarayanan, Ramarathnam Venkatesan,Prasad Naldurg, Sean and Adithya Hemakumar, SPACE: Secure Protocol for Address-Book based Connection Establishment, Fifth Workshop on Hot Topics in Networks (HotNets-V) (Nov 29 & 30, 2006, in Irvine, California).

[14] William Stallings, Cryptography and Network Security, Principles and Practices, Third Edition.

[15] Hansmann, Nicklous, Shaeck, Schneider and Seliger, Smartcard Application Development using Java, Second Edition.