Mitigating Blind Signature Attacks Using Daily Updated Quotes (DUQ)

S.Kalaiselvi, B.Sathya (PG Student), Ms. A. Priyadharshini, (Assistant Professor)
Computer Science and Engineering,
Knowledge Institute of Technology, Salem,Tamilnadu,India.
Email Id: kalaislv12@gmail.com, sathyacse24@gmail.com, apcse@kiot.ac.in.

Abstract **--** To ensure integrity of data, digital signature scheme is used.Blind signature is a popular approach where a third party is used to sign the message. The blind signature scheme is more vulnerable to attack. If the signature is compromised then the whole system is exposed to the attacker. In the previous digital signature schemes chakraborty-melta signature scheme is used to generate the blind signature. But the stamp of the signature "z" in the algorithm is easy to compute. So there is possiblity for two attacks. Hence, we propose Daily Updated Quotes(DUQ) Scheme that makes the stamp of the signature diffcult to find by the attacker. Hence, the blind signature is secured for further processing.

Keywords*: Blind signature, Elliptic curve, Stamp*

*****

1. INTRODUCTION

Blind signature scheme is introduced by David Chaum [3].In a blind signature scheme the content of a message is blinded before it is signed.

The user sends a message to signer. Then the signer verifies the message and sends back to the user. After unbinding, the message gives the blind signature and blind message.

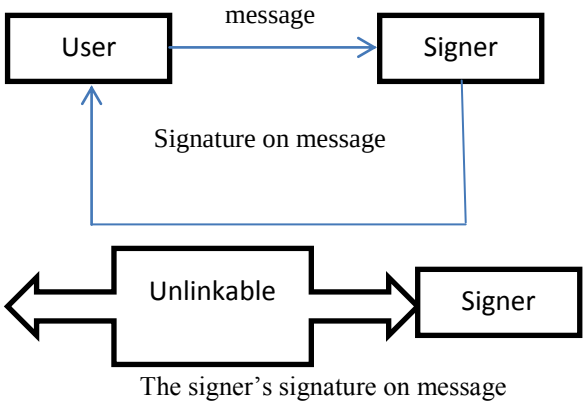

Fig 1.Blind signature

In general signature generation and verification scheme the user sends the message to the signature verifier. Message will be blinded and attacker will crack the message by using key that he have. Once the message is attacked then it is a readable message. Finally verifier will verify whether it is true or false. A secure blind signature scheme satisfies following three properties:

1) Un-linkable: Attacker cannot get exact message from the user.
2) Randomized: Enciphers message through randomly choosing cipher text and it will give the greater security and it eliminate plaintext attacks.
3) Message recoverable: when network or computer failure occurs it would not interrupt the message and it will surely deliver.

Because of the un-linkable and randomized properties blind signature scheme has been working expansively in confidentiality oriented applications, such as e-voting system [5, 8].

Recently, Chakrabortyand Mehta [2] projected blind signature security established on the elliptic curve discrete logarithm problem. In this paper, easily A (y,z) can able to copy and use signature on any message, and after getting a signature from a signer, A also know the secret of the signer. Therefore, Chakraborty and Mehta blind signature scheme cannot be used in e-voting system because it is not secure.

2. RELATED WORK

2.1 Elliptic curve group

Let n>3 be a large prime and F_n be a finite field. An elliptic curve H over F_n is the set of all points A=(y,z) that satisfy the equation

$$z^2 = y^3 + sy + t(\mathrm{mod} n)$$

Here s,t € F_n are constants such that $4s^2 + 27t^2 \neq 0$, together with an infinity point T. the elliptic curve H can form a cyclic group under the point addition operation C=A+B which is defined according to a chord-and-tangent rule. Particularly, we define x. P=P+P+…..+P(x times).

2.2 General security notations of digital signature scheme

As mentioned previously, a digital signature scheme must satisfy message recoverable. Here, we list all type of forgeries of signature schemes:

1) **Universal forgery:** The creation of a valid signature for any given message. An adversary capable of universal forgery is able to sign messages which he chose himself, messages chosen at random or even specific messages provided by an opponent.
2) **Selective forgery :** The creation of a message/signature pair. Where message has been chosen by the adversary prior to the attack.
3) **Existential forgery:** The creation of at least one message/signature pair.The adversary need not have any control over message.

2.3 Review of Chakraborty-mehta blind scheme

In this section, we briefly review Chakraborty-Mehtablind signature scheme.
Let H be a elliptic curve group and A be a generator of H with order b. The Chakraborty-Mehta blind signature scheme run as follows.
Setup: Input H, A and b, the algorithm outputs a cryptographic hash function.
H:{0,1}*→Z_b^* The signer choose y €Z_b^*uniformly at random and sets (B=yA,y) as its public/private secret key pair.
Blind: The user with message M wants to get a blind signature on M. He first computes k= h (M) and d=kB, then sends d to the signer.
Sign: On input d and the secret key y, the signer does the following steps:

1) Compute d′=y^{-1}d=kA.
2) Generate a stamp of the blind signature w= [nounce||date||place quote of the day].
3) Compute C=d′+ h (w)P and v=y - h(w).
4) Output the signature (C,v,w).

Verify: Input is the message M and the signature(C,v,w), the verifier accepts the signature if and onlyif the equality v(A)-B+C=h(m)A holds.
The correctness of the scheme can be verified as follows.
v(A) -B+C
= (y-h (w))A-B+d′+h(w)A
=yA-B+d′
=d′
=h(M)A

2.4 Cryptanalysis of Chakraborty-mehta signature scheme

2.4.1 Attack 1

If the malicious user A needs to get the signer's secret key y, then the attacker can do the following steps:

1) A request a blind signature on a message M. Then the signer will compute and output signature (C,v,w) as a response.
2) After receiving, A computes h (w).
3) Finally, the secrete key is obtained as y=v+ h (w).

The attack 1 is successful.we know that the algorithm 'sign' (v=y-h (w)). The signer secret key y will be able to generate valid signature of any messages.

If h (w) is known then the secret key will be found.

2.4.2 Attack 2

If the mean user wants to get message M with valid signature then, the following steps will be followed:

1) $\mathcal{A}$ first produce the signature w= (nounce|| date|| place)
2) Then $\mathcal{A}$ compute h(w) and h(M).
3) $\mathcal{A}$ selects v£Z_q^* equally at random.
4) Finally, $\mathcal{A}$ computes C=h(M)A+B-v(A), where B is the signer's public key.

The copied signature on M is (C,v,w).

We can see that v (A)-B+C=v(A)-B+ h(M)A+B-v(A)=h(M)A

Therefore the signature (C,v,w) will pass the verifier proving. That the signature (C,v,w) copied by $\mathcal{A}$ is valid.

3. PROMOTED WORK

3.1 Daily Updated Quotes(DUQ) Scheme

3.1.1 Selecting expiration dates and using sub keys

By default, a DSA master signing key and an ElGamal encryption sub-key are generated when a new key-pair is created. This is convenient, because the roles of the two keys are different, therefore the keys need to have different lifetimes. The master signing key is used to make digital signatures, and it also collects the signatures of others who have confirmed the identity. The encryption key is used only for encrypting and decrypting the encrypted documents

sent to the receiver. To maintain the digital signature for long period, a key needed to be framed too hard. On the other hand, the encryption sub-key may be changed periodically for extra security, since if an encryption key is broken, the attacker can read all documents encrypted by that key both in the future and from the past.

In most of the cases we would not use the master key to expire and it has two reasons for choosing an expiration date. First, you may intend the key to have a limited lifetime. For example, it is being used for an event such as a political campaign and will no longer be useful after the campaign is over. Another reason is that if you loss control of the key and do not have a revocation certificate with which to revoke the key, having an expiration date on the master key ensures that the key will eventually fall into disuse.

Changing encryption sub-keys is straight forward but with be inconvenient. If you generate a new key-pair with expiration date on the sub-key, that sub-key will eventually expire. Shortly before the expiration you will add a new sub-key and publish your updated public key. Once the sub-key expires, those who wish to correspond with you must find your updated key since they will no longer be able to encrypt the expired key. This may be inconvenient depending on how you distribute the key. Since no extra signatures are necessary the new sub-key will be signed with your master signing key, which presumably has already been validated by your correspondents.

The inconvenience may or may not be worth the extra security. Just as you can, an attacker can still read all documents encrypted to an expired sub-key. Changing sub-keys only protects future documents. In order to read documents encrypted to the new sub-key, the attacker would need to mount a new attack using whatever techniques he used against you the first time.

Finally, it only makes sense to have one valid encryption sub-key on a key-ring. There is no additional security gained by having two or more active sub-keys. There may of course be any number of expired keys on a key-ring so that documents encrypted in the past may still be decrypted, but only one sub-key need to be active at any given time.

3.1.2 HASH FUNCTION

A hash function maps an input a variable length message into a fixed length value, which is called as message digest or hash value.

A document's digital signature is the result of applying a hash function to the document. To be useful, however, the hash function needs to satisfy two important properties. First, it should be hard to find two documents that hash to the same value. Second, given a hash value it should be hard to recover the documents that have produced that value.

Some public-key ciphers could be used to sign documents. The signer encrypts the document with his private key. Anybody who is willing to check the signatures can that see the document simply uses the signer's public key to decrypt the document. This algorithm satisfies the two properties needed for a good hash function, but in practice, this algorithm is too slow to be useful.

An alternative is to use hash functions designed to satisfy these two important properties.

SHA-512(Secure Hash Algorithm)

The SHA loas developed by NIST& published as a federal information processing standard(FIPS 180) in 1993, a revised version was issued as FIPS 180-1 in 19952 is generally to as SHA-1.

SHA-512 is the version of SHA with a 512-bitmessage digest. The algorithm takes as input a message with a maximum length of less than 2^{128} bits and produces as output a 512 bit message digest .The input is processed in 1024 bit blocks.

MD5 (Message Digest Version 5)

Using such an algorithm, a document is signed by hashing it, and the hash value is the signature. Another person can check the signature by also hashing their copy of the document and comparing the hash value they got with the hash value of the original document. If the two hash values match, it is almost certain that the documents are identical.

MD5 has been developed by Ron Rivest .It provide more security MD5 has been Developed. The algorithm takes as input a message of 512-bit blocks& produces as output a 128-bit message digest.

4. CONCLUSION

Digital signature scheme is more powerful authentication mechanism and it also addresses integrity problems. But blind signature scheme is vulnerable to attack. We proposed a Daily Updated Quotes(DUQ) scheme which makes the Stamp “z” of the signature more secure as

 ISSN: 2321-8169
1495 – 1498

the stamp is encrypted. Hence the blind signature is free from attacks.

REFERENCES

[1] J. Camenisch, J. Piveteau, and M. Stadler, Blind signatures based on the discrete logarithm problem," in Eurocrypt '94, LNCS 950, pp. 428-432, Springer-Verlag, 1995.

[2] K. Chakraborty and J. Mehta, A stamped blind signature scheme based on elliptic curve discrete logarithm problem," International Journal of Network Security, vol. 14, no. 6, pp. 316-319, 2012.

[3] D. Chaum, "Blind signatures for untraceable payments," in Crypto '82, pp. 199-203, 1983.

[4] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen message attacks," SIAM Journal of Computing, vol. 17, no. 2, pp. 281-308, 1988.

[5] I. Lin, M. Hwang, and C. Chang, "Security enhancement for anonymous secure e-voting over a network," Computer Standards and Interfaces, vol. 25, no. 2, pp. 131-139, 2003.

[6] T. Okamoto, "Efficient blind and partially blind signatures without random oracles,"in The 3rd Theory of Cryptography Conference, LNCS 3876, pp. 80-99, Springer-Verlag, 2006.

[7] M. Ruckert, "Lattice-based blind signatures," in Asi-acrypt '10, LNCS 6477, pp. 413-430, Springer-Verlag, 2010.

[8] F. Rodrguez-Henrquez, D. Ortiz-Arroyo, and C.Garca-Zamora, Yet another improvement over the Mu-Varadharajan e-voting protocol," Computer Standards and Interfaces, vol. 29, no. 4, pp. 471-480, 2007.