International Journal of Soft Computing 2 (1): 113-117, 2007

Two Efficient Digital Multisignature Schemes

Sattar J. Aboud
Department of Computer Science, The University for Graduate Studies,
Faculty of IT, Amman, Jordan

Abstract: In this study we introduce two methods of revising the RSA algorithm to allow multisignatures being simply applied by set of users. The proposed algorithms will be suitable for employ in organizations. However, general uses might be the signing of cheques for e-funds transfer where two or more representatives are needed to sign the cheque. The algorithms suggested in this paper are not restricted to RSA scheme. It means that every encryption algorithm with the multiplicative feature will work. For reality, however, we will employ RSA widely in this study.

Key words: RSA scheme, digital multisignature, e-funds transfer

INTRODUCTION

The concept of digital signature is now common as a way of replacing hand written signatures in e-communications[1]. In many uses a cheque may require to be signed by more than one entity[2-4]. So when a signature needs more than one entity is usually called a multisignature.

One technique of implementing a digital multisignature is by employing a so called threshold scheme[2,3] which distributes knowledge of the singing key amongst many entities. However, in this example all the signers should set together at the same time and at the same place for redesign the key, which is obviously irrelevant in various uses. Additional possibility is subsequent signing by the different entities, when the signature is checked by deleting every signature in turn. This has some drawbacks such as the time required for verification is repeated by the number of signers[4]. Also the verifier will require the logical verification information for every signer participated[5]. In addition, in the RSA signature scheme, which is the most known scheme, re-blocking being important when a successive signer has an expanded modulus[6].

The technique of performing a multisignature algorithm employ RSA scheme was described in[7]. This was substantially a technique of defeating the relocking difficulty stated above by specifying an individual modulus corresponding to authority supremacy; so that a superior is always own a larger modulus than those on his hand. This algorithm is just suitable when the order of signing is predetermined. The concept behind the schemes proposed is to extend the RSA scheme to a multi key encryption. A certain technique of achieving multi key encryptions the keys should be inserted at the same time[8,9]. The algorithms proposed will employ the multiplicative ability of inverse[10-13] to create a method when it is needed.

In 1978 RSA suggested novel scheme for public key encryption, its security relied on the difficulty of integer factorization[7]. The implementation of RSA scheme for multiple signings of a determined message produces bit expansion problem intrinsically. The early algorithms that face this difficulty are re-blocking algorithm[14], repeated square and multiple algorithm[15,16] and fast exponentiation algorithm[17].

Another idea is suggested for a multisignature scheme in which many signers creates a digital signature for a determined message[18]. To face the problem of bit expansion in the RSA multisignature, they permitted the signer to have the RSA modulus with a special bit length corresponding to his place in an orderly structure. So, the signing order is limited.

Alternatively, another suggested multisignature algorithm with out limitation of the signing order[19,20]. In this algorithm, when the size of median signature surpasses a pre-determined threshold value, then the increment bits surpassing the threshold value are added to the document. Thus the size of expanded message relies on the number of signers and the bit length of every signer's RSA modulus.

There are another two suggested multisignature algorithms with out bit expansion[21]. In the first scheme the signing order is selected corresponding to the length of signers' exponent key. The second algorithm is relied on re-encryption scheme with permutation polynomials technique. Although their, multisignature algorithms have no bit expansion difficulty and the signing order is not

limited, every signer has the RSA modulus with the same bit length and the computational requirements of getting the multisignature is expanded.

In this study we present two methods of altering the RSA algorithm to enable multisignatures to be simply used by many entities. The proposed algorithms will be suitable for employ in organizations. However, general applications can be the signing of cheques for e-funds transfer where two or more organization representatives are needed to sign the cheque. The algorithms suggested in this study are not restricted to RSA scheme. It means that every encryption method with the multiplicative property will effect. For concreteness, however, we will employ RSA widely in this study.

THE FIRST SUGGESTED DIGITAL MULTISIGNATURE SCHEME

The algorithm proposed allows two entities to sign a cheque which can be validated by another entity. The concept is to extend the RSA algorithm by owning three keys instead of two, two secret keys and one public key. The authority selects a modulus n which is the product of two large prime numbers similar to the RSA algorithm. The two secret keys are then selected randomly in the interval [1, n-1], according to the condition that they are relatively prime to θ (n). Assume that these keys x and y. The public key e is then selected so that: $x^* y^* e^* \equiv 1 \bmod \theta(n)$ x and y are now send to the authorized signers and e is publish publicly. In order to sign the cheque c the first signer computes:

$$s_1 = c^x \bmod n$$

and send s_1 to the second signer. The second signer can now recover c from s_1 in order to see what he is to sign by:

$$c = s_1^{(y^* e)} \bmod n$$

since he knows both y and e. If he satisfied he now signs s_1 to computes:

$$s_2 = s_1^y \bmod n$$

and send s_2 to the receiver. Since e is public, the receiver can check the validity of the cheque by computing: $c = s_2^e \bmod n$

The cheque should be signed by the two authorized signers in order to computes s_2. The order of signing in this algorithm is not important.

Example: Suppose n = 187. , which is the product of (p = 511, q = 17), then θ (n) = 170. Assume we select x = 7 and y = 13, then the , gcd (7, 170) + 1, also the gcd (13, 170) 1. Suppose that we find the public key e = 71 as follow:

$$7^* 13^* 71 \equiv 1 \bmod 170$$
$$6461 \equiv 1 \bmod 170$$

In order to sign the cheque c = 12, the first signer computes:

$$s_1 = 12^7 \bmod 187 = 177$$

And send s_2 to the second signer. The second signer can now recover c from s_1 as follows:

$$c = 177^{(13^*71)} \bmod 187 = 133$$

The second signer sign s_1 to computes:

$$s_2 = 177^{13} \bmod 187 = 23$$

The second signer now sends s_2 to the receiver. The receiver can check the validity of the cheque by computing:

$$c = 23^{71} \bmod 187 = 23$$

Knowing of s_1 is of no use to an opponent. It appears clear that forging a cheque for this algorithm is similar to forging the RSA signature with private key (x*, y) and in this meaning the algorithm is similarly secure as the RSA scheme. However, from the key management point view it is more secure to divide the key into two divisions and hold them individually. However, there are various uses where distributing the responsibility for authorization between two persons is needed.

The multiplicative characteristic used in this algorithm can also be employed to attack RSA signature in some situations[22]. For instance, since $(m_1^* m_2^x) = (m_1, m_2)^x$ the signature of m_1, m_2 can be figured out from those of m_1 and m_2. Various tools are available to keep away from these attacks, and they are also usable in this algorithm. One technique is to employ a one way hash function h in which the message should be hashed pre-signing, To avoid the potentially of collision of messages, it is preferable that the hash function employed must have a 160 bits with Secure Hash Algorithm[23]. So that $h(m_1, m_2) = h(m_1) * h(m_2)$ This also has the benefit that just single block requires to be signed.

THE SECOND SUGGESTED DIGITAL MULTISIGNATURE SCHEME

In this study we introduce a second algorithm to solve the difficulty of multisignature with more than two signers. Though, it needs every signer to sign individually and the product is combined which could be not convenient in numerous uses. In this algorithm the publishing organization again chooses the private keys arbitrarily. Assume that there are three signers wanted r_1, r_2 and r_3 are chosen randomly and this time the exponent key e is computed as follows:

$$(r_1+r_2+r_3)^* e \equiv 1 \bmod \theta (n)$$

Every signer i determines the message m and signs it to find:

$$S_i = m^{ri} \bmod n$$

The three signed copies are then multiplied by certain main office to find:

$$S = S_1 * S_2 * S_3 \bmod n$$

Then this will sending to the receiver. The receiver and any participant of the public can check the signature employing e as follows:

$$s^e \bmod n = (S_1 * S_2 * S_3) \bmod$$
$$= m^{[(r_1+r_2+r_3)*e]} \bmod n$$
$$= m$$

Example: Suppose that $r_1 = 11$, $r_2 = 13$, $r_3 = 19$. Assume e = 19 then (11+13+19) *19≡ 1 mod 48. Suppose m_1 12, $m_2 = 21$, $m_3 = 8$. So

$$s_1 = 12^{11} \bmod 65 = 38$$
$$s_2 = 12^{13} \bmod 65 = 12$$
$$s_3 = 12^{19} \bmod 65 = 38$$
$$s = 38 * 12 * 38 \bmod 65 = 38$$

To verify the signature using e as follows:

$$38^{19} \bmod 65 \equiv (38 * 12 * 38)^{19} \bmod 65$$
$$12 \equiv 12$$

This scheme can be enhanced to allow any number of signers to be included.

Digital multisignature scheme which requires the knowledge of the message as an input to the verification algorithm is called digital signature scheme with appendix. Digital signature schemes with appendix are the most commonly employed in practice[24]. They based on cryptography hash function rather than customerized redundancy functions, and are less prone to existential forgery attacks. The proposed schemes are designed to use a combination of an appropriate one way hash function h with which m shell be hashed before signing in order to bound the size of the key in verification. To avoids the potentiality of the message collisions. It is preferable that the hash function employed must have a 160 bits product and Secure Hash Algorithm[25, 26], which seems to be appropriate choice.

ENHANCING THE SCHEMES

In some status the number of signers required could be numerous. For instance in an authority any two of a member of authorized signers can be accepted[24]. So according to this point the above algorithm could be extended to meet this situation. However, when there are n possible signers then n random private keys, r_1, r_2,....$r_n$are selected. The public key e is selected so that:

$$r_1, r_2, \ldots\ldots r_n * e \equiv 1 \bmod \theta (n)$$

Every signer is then given all the secret keys unless one. For instance the j^{th} signer is provided all r_i unless r_j . Every signer keeps all these keys and also their result. Suppose $r_j = r_1 \ldots r_{(j-1)}\, r_{(j+1)} \ldots r_n$. If the j^{th} signer wants to sign a cheque che signs it to compute:

$$s_1 = c^{rj} \bmod n$$

and attaches his identity. The other signer can then achieve the signing by looking up the nonexistent key, which also permits him to check the cheque and then to compute:

$$s_2 = s_1^{rj}$$

The receiver and any one of the people can again check the signature by decryption with e.

SECURITY OF THE SCHEMES

Many useful multisignature schemes does not take any proof of security[27-29], it is known that breaking RSA signature is based on the factorization difficulty. The only visible attacks on the suggested schemes are as difficult as factoring the modulus n, but it is not shown if there is any certain efficient attack. The following lemmas are simply verified:

Proposition 1: If messages m_1 and m_2 are congruent mod m, then the value s ($0< s<n$) is a right signature of m_1 if and only if s is a valid signature for m_2.

Proposition 2: Values s_1 and s_2 are right signatures for the same message m if and only if s_1 and s_2 are congruent mod n .

Both these answers disclose the picture of the signature space of the scheme. Many practical signature schemes including the RSA are tend to existential forgery attacks if a hash function is not employed pre-signing[30-32]. In such attacks, there are unrestricted numbers of signatures for random messages can be created. For the suggested algorithm a simple existential forgery is that the value s = 1 is the signature for the message m = c. Additional random signatures appear difficult to perform.

Choosing forgery relates to the complexity of forging a signature of a message selected in advance by the opponent. With employ of one way hash functions this seems the only technique to obtain any valid signature[33,34]. The opponent selects a message m and is needed to compute a signature s with $s^m \bmod n = c$. The possibility to calculate the signature s from knowledge of the public key is the identical to breaking the RSA encryption method for a known cipher text c and public key e, with the part data that there is a factor of θ (N) for size 160 bits. It is unobvious if the part data is aid in factoring n. This is similarity to the security of identification protocol in[35].

A suitable selected message attack employ of signatures on selected messages and is difficult to hinder than attack employing just the public key. Such an attack is no longer identical to an attack on RSA but could accord to a status where an opponent could select the public key of the RSA and get the original message according to the cipher text c. It dose not seem to be any clear approach that this aids an opponent.

CONCLUSION

Two new digital multisignature schemes are suggested depend on well known mathematical background but employing a new technique. The algorithms seem to be secure in comparison with the well known algorithms, although proves of security have been done. Additionally the algorithms offer some specialties, which may demonstrate advantages. Both the multisignature schemes are stabilized in the meaning that public and private key calculations are likely equal. Also, the average computational conditions for multisignature generation and verification are similar to the RSA algorithm. It is interesting to take account of protocols in which the new multisignature may be employed as primitive. There is also important similarities are found in elliptic curves and other classifications.

RERERENCES

1. Sattar, J.A. and A. El Sheikh, 2004. A new method for public key cryptosystem and digital signature scheme based on both integer factorizations and discrete logarithms. Asian J. Inform. Tech., 4: 284-289.
2. Shamir, 1979. How to share secret. Communication of the ACM, pp: 612-613.
3. Desmedt, Y.G., 1994. Threshold cryptography. European Transactions on Telecommunication, 5: 449-459
4. Fiat and A. Shamir, 1986. How to prove yourself: Practical solutions to identification and signature problems. CRYPTO '86 (LNCS 283), pp: 186-194.
5. Ammar, M. and M. Mousbah, 2002. A High efficiency method for automatic signature verification. Patent No. 09/453730, USA.
6. Kiesler, T. and L. Harn, 1990. RSA blocking and multisignature schemes with no bit expansion. Electronic Lett., pp: 1490-1491.
7. Rivest, R., A. Shamir and L. Adelman, 1978. A Method for obtaining digital signatures and public key cryptosystems. Communication of the ACM 21, pp: 120-126.
8. Carrol, J., 1994. The resurrection of multiple key ciphers. Cryptology.
9. Beaver, D., 1991. Secure multi-party protocols and zero knowledge proof systems tolerating a faculty minority. J. Cryptol., 4: 75-122.
10. Stein, J., 1967. Calculating the multiplicative inverse. Com. Phys., 1: 397-405.
11. Sattar, J.A., 2004. Baghdad method for calculating Multiplicative Inverse. international Conference on Information Technology, Las Vegas, Nevada, USA, pp: 816-819.
12. Sattar, A., 2005. Fraction-Integer Method (FIM) for calculating multiplicative inverse. J. Sys. Cybernetics and Informatics, USA, pp: 5.
13. Jordan, J., 1989. Fast multiplicative inverse to modular arithmetic. Cryptography and Coding, Clarendon Press, Oxford, pp: 369-379.
14. Kohnfelder, L., 1978. On the signature re-blocking problem in public key cryptography. Communication of the ACM, 1: 179.
15. Levine, J. and J. Brawley,1997. Some cryptographic applications of permutation polynomials. Cryptologia, 1: 76-92.

16. Knuth, D., 1996. The Art of Computer Programming Semi numerical Algorithms 3rd (Ed.), Addison Wesley, pp: 319-339.
17. Bricell, B.F., D.M. Gordon, K.S. McCurley and D.B. Wilson, 1992. Fast Exponentiation with Pre-Computation, In Advances in Cryptology-Proceedings Europcrypt, pp: 200-207.
18. Itakura, k. and K. Kakamura, 1983. A Public Key Cryptosystem Suitable for digital Signatures. NCE J. Res. Dev., pp: 71.
19. Okamoto, T., 1998. A digital multisignature scheme using bijective public key cryptosystems. ACM Transaction, Computer Systems, 6: 432-441.
20. Gennaro, R. and V. Shoup, 1998. Securing threshold cryptosystems against chosen cipher text attack. Europcrypt, pp: 1-16.
21. Harn, L. and T. Kiesler. New scheme for digital signatures. Electronics Lett., 25: 1527-1528.
22. Evon, M., Abu-Taieh and S.J. Aboud, 2003. A New Factoring Algorithm. International Conference on Security and Management SAM' 03, Las Vegas, Nevada USA, pp: 341-347.
23. FIPA, 180-1, 1995. Secure Hash Standard. US Department of Commerce/NIST.
24. Menezenes, P.V.O. and S. Vanstone, 1997. Handbook of Applied Cryptography. ARC Press.
25. FIPA, 180-1, 1994. Secure Hash Standard. US Department of Commerce/NIST.
26. Musbah, M., Aqel and M. Ammar, 2005. Function structure and operation of a modern system for authentication of signatures of bank checks. Pak. Infor. Tech. J., 4: 96-105.
27. Chiou, S. and C. Laih, 2004. On the implementation of (2, n) audio cryptography schemes without computing devices. Intl. J. Elect. Engin., 11: 53-58.
28. Laih, C. and S. Chiou, 2003. Cryptanalysis of an optimized protocol for mobile network authentication and security. Inform. Proc. Lett. 6: 339-341.
29. Grieu, F., 2000. A chosen messages attack on the ISO/IEC 9796-1 Signature scheme. Eurocrypt LNCS, pp: 1807-70ff.
30. He, W., 2001. Digital signature scheme based on factoring and discrete logarithms. Electronics Lett., 37: 220-222.
31. Chiou, S. and C. Laih, 2003. A tempo-based t-out-of-n audio cryptography scheme. IEICE Transactions on Fundamental of Electronic, Communication and Computer Sciences, 8: 2091-2098.
32. Sun, H., 2002. Cryptanalysis of a digital signature scheme based on factoring and discrete logarithms. NCS.
33. Douglas, R.S., 2006. Cryptography theory and practice. CRC 3rd, pp: 117-149.
34. Arto, Salomaa, 1996. Public Key Cryptography. 2nd (Edn.), Springer.
35. Brickel, E. and K. McCurley, 19962. An interactive identification scheme based on discrete logarithms and factoring. J. Cryptol., 5: 29-39.