Asian Journal of Information Technology 5 (12): 1361-1366, 2006

Robust Multiplesigncrypt Scheme with Order Flexiblity

[1]Paul Rodriques, [2]A. Chandrasekar, [3]S. Radhika and [4]N.V. Ballaji
[1]A.k. College of Engineering, [2]CSE St.Joseph's College of Engineering
[3]Sathyabama University, [4]CSE- St.Joseph's College of Engineering

Abstract: In everyday life many documents require authorization from more than one party. For instance contracts, decisions and workflow systems etc. The purpose and use of multiple signatures are various. Multiple signature schemes realize that plural users generate the signature on a message and that the signature is verified. There is many multiple signature protocols available which satisfy order verifiability but not order flexibility and a message requiring N signatures would be increased in length by about N times as much as for one signature which is not feasible in economic perspective. An existing scheme does not provide security from unauthenticated message. In this study we propose a new multiple signature method with the following features. Multiple signatures with the same length as a single one. Multiplesigncrypt: Enhancing security by combining multiple signatures with encrypted data. Robust: Preventing unauthenticated users from attacking the text. Order Flexibility: Allowing the order of signers to be random. We have shown a RSA-based multisignature scheme which satisfies the above mentioned features.

Key words: Authorization, signature, flexiblity, perspective

INTRODUCTION

A few centuries ago, signature and eventually a wax seal were the only way to certify the authentication of a document. Since that time and until today, when a signature is apposed to a treaty by a president or to the wedding license by a happy couple, it is assumed that[1]:

- The signature binds the signer to whatever the document states.
- The document will not be changed once the parties have signed it.
- A signature on one document will not be transferred fraudulently to another.

It is a challenge to make these assumptions respected by the electronic equivalent of the traditional handwritten signature: the Electronic signature, or E-signature[2].

E-signatures are generally divided into two separate categories: digital signatures and electronic signatures. In contrast with digital signatures, electronic signatures do not rely on cryptographic methods and are often biometrics-based solutions. Digital signatures can be classified into two main categories: single signature and Multiple signature (or multisignature). Single signature refers to the cases where only one party signs a document, while multiple signatures refers to the cases where more than one party sign a single document[3].

Suppose that a group of people wish to send a message so that the receiver knows for certain that the message came from those specific people and can later prove this to an impartial third party[4]. For example, the message might be a contract requiring the signatures of several corporate officers, or a legal document requiring witnesses and notarization. The procedure for producing the signature should guarantee that nobody else can forge the signature or modify the signed message and that no signer can later deny signing. Even if another, possibly larger, group of corporate officers wanted to change the message, or masquerade as the actual signers, they should be unable to do that unless the group included all of the actual signers. One possible way to achieve this would be to have each party append a separate digital signature to the message. This means a message requiring n signatures would be increased in length by about n times as much as for one signature[5]. For economy, it would be preferable to have a joint signature about the same length as a single signature, provided that it does not take significantly more execution time to produce it than n separate signatures.

In RSA cryptosystem, it takes less time to produce a joint signature than N separate signatures. In proportion as the spread of personal computers and network, messages like documents, data, software etc have been circulated through internet. In such environment, an entity sends/forwards an original message to others or

Corresponding Author: A. Chandrasekar, Department of CSE, Research Scholar-Annauniversity, St. Joseph's College of Engineering, Chennai-119

sends a modified message to others. Through the process of circulation, keeping track of or maintaining the order in which the person has to sign becomes a tedious task. Recently, it has been a new problem for computer virus to be mixed into a message through this circulation. Apparently, it is an obstacle to circulate messages soundly through internet. This is why; a multisignature scheme suitable for such an environment is required.

Definitions

Robustness: If the signature verification fails, then we prevent such an unauthenticated message from damaging the receiver. By employing this, our scheme becomes more robust.

Multiplesigncrypt: Encrypted multisignature which has a feature that a message cannot be recovered if the signature verification fails.

Order flexibility: Neither order of signers not signers themselves need to be designated beforehand. Therefore, we can easily change order of signers.

Digital signature: The digital signature category is the most secure and most full-featured type of signature. It relies on Public Key Cryptography (PKC). Different PKC schemes have been used to implement digital signature and data encryption[3]. For example:

- The RSA (Rivest-Shamir-Adleman) scheme,
- The Digital Signature Algorithm (DSA) scheme,
- The ElGamal scheme,
- The Elliptic Curve Digital Signature Algorithm (ECDSA) scheme

When using these schemes to implement digital signature, a pair of mathematically related keys are involved: A private key and a corresponding public key. Public keys are published and can be stored in directories. Private keys must be kept secret and only known by the user and so are usually stored on encrypted portions of a hard drive, on Smart Cards or stored on a network and delivered only after the appropriate Password is entered. The algorithms used are asymmetric. This key system obeys to these mathematical properties[6]:

- Encrypting a message with a private key and then decrypting the result with the corresponding public key, will restore the initial message.
    - Given a public key, it is not possible to find out the corresponding private key shown in Fig. 1.

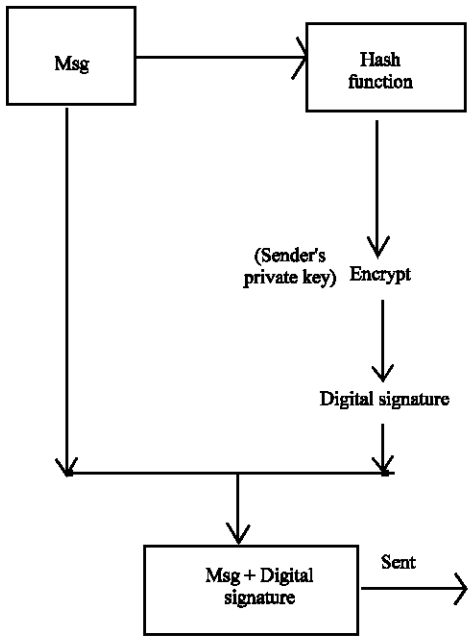

Fig. 1: A single user signing a message[7]

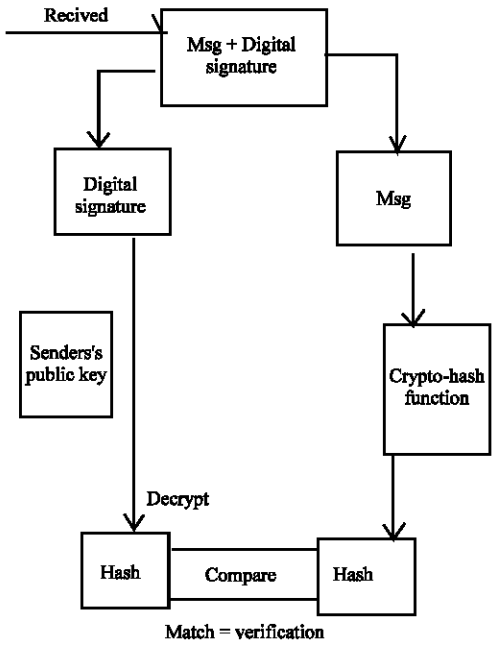

Fig. 2: Receiving a signed message[7]

The Fig. 2 illustrates the steps followed by the receiver of the signed message.

The encrypted hash that came with the message is decrypted using the sender's public key and the result is compared with the hash generated by the receiver from the message itself. If they match, then the message can be

considered as authentic. If the hashes don't match, this can be a sign of message modification, transmission error or identity usurpation[8,9].

This digital signature scheme guarantees three information security properties:

Authentication: The signer is well identified by the private/public key relation.

Non-repudiation: The signing party cannot later on deny performing the action, since the private key was used for encryption process. Note that if a *Symmetric key* cryptography was used, the non-repudiation properties could not be guaranteed.

Integrity: Since the signature itself is associated to the content to the message, any message alteration would make the signature invalid.

A secure hash: The basis for the digital signature is a secure hashing function, or message digest. This means a hash function where any change Changing any block of the message will almost certainly change the value of the hash. If the prime factors p and q of N are chosen so that either p or q or both are = 3(mod 4) then changing any one block of the message will step the value of H through half of the values 0 to N-1 twice each (plus one additional value that occurs only once). This is because one quarter of the values from 1 to N-1 is quadratic residues and one quarter is of the form (N-quadratic residue).

In the study[9], the hashing modulus N was the sender's modulus. This choice meant that nobody except the sender could create a message which has the same hash value as the genuine message. That was perfectly secure in a single-sender single-signature situation. However, here we have multiple signatories[2], each of whom is responsible for the contents of the message. We can use neither a sender's modulus, nor the receiver's modulus, for this would allow that person to substitute another message for the genuine message.

I propose that the cryptosystem have an additional public modulus specifically for the purpose of constructing this hash. The to message almost certainly changes the hash value and nobody can modify a message in such a way that the hash value remains unchanged. A suitable hash function is described in[4]. Let the message M consists of the blocks B_1, B_2... B_k. The message can be hashed in the following way:

$$H_0 = Q,$$
$$H_i = D \cdot H_{i-1} + J_i \pmod N,$$
$$H = H_K$$

where Q is some fixed large number used as a seed, D is a fixed multiplier and J_i is defined as

$$J_i = B_i^2 \text{ if } B_i < N/2,$$
$$J_i = N - B_i^2 \text{ if } B_i > N/2.$$

Since N is odd, B_i cannot equal N/2. If N has b bits, then each message block B_i will be b-1 bits long, except possibly the last block.

Factorization of this modulus would be known only to some trusted central authority, or perhaps to nobody at all. This could be arranged by using a computer program that generated two large primes and reported their product, but not the primes themselves. The choice of these primes could depend on physical events that could never be replicated, such as the exact timing of keystrokes from several mutually isolated keyboards.

Since this master modulus or master key will be the basis for signed documents throughout the communications network, anyone who factored the master key could alter documents at will. To assure this does not happen, the master key must be chosen to make factoring difficult beyond anything that can be achieved today or in the foreseeable future. I suggest that the master key N be the product of two primes p and q satisfying $p>10^{100}$, $q>10^{100}$ and abs $(p-q)>10^{50}N^{1/4}$. They should be of the form $p=2p'+1$ and $q=2q'+1$ with p' and q' prime. A convenient size for the product N=pq would be 1024 bits, corresponding to 308 to 309 decimal digits. As computers get faster and factoring algorithms improve, this size should be revised upwards to stay at least 5-10 years ahead of the state of the art.

Notations used: Before looking at the multisignature scheme, we define the following notations.

An original message M_1 is given by I_1. $M_{1,2,...i}(i > 2)$, denotes a message which is added some modification by the i-th signer Ii. The difference between $M_{1,2,...,i-1}$ and $M_{1,2,...,i}$, which means the modification by Ii, is defined as,

$$m_i = \text{Diff}(M_{1,2,...,i-1}, M_{1,2,...,i}).$$

We also define a function Patch which recovers a message,

$$M_{1,2,...,i} = \text{Patch}(m1, m2, ..., mi).$$

For the sake of convenience, we denote m1 =Patch (M_1). We use a signature scheme with a message recovery feature. The signature generation or message recovery function is denoted by[10].

Sign $(sk_i, m_i) = sgn_i$, or $Rec(pk, sgn) = m$, respectively,

Where ski is Ii's secret key and pk_i is Ii's public key. Let h1 be a hash function and ID_i be signer's identity information. We also use two operations * and ? in a group G

$$(A^* B) \ddot{O} B = A \ (!A,B \in G).$$

For example in case of G = Zp, * and $\ddot{O}$ mean modular multiplication and modular inversion, respectively.

Rsa-based multisignature scheme: An authenticated center publishes small primes in addition to {1}, {rl} = {1, 2, 3, 5, • • •}. Assigner Ii with identity information IDi generates two large primes pi and qi secretly and computes public keys ni,l and ei,l, Zni,l in such a way that

$$n_{i,1} = p_i q_i r_l, \ (l \geq 1)$$
$$L_{i,l} = LCM((p_i - 1), (q_i - 1)) \qquad (l = 1)$$
$$LCM((p_i - 1),(q_i - 1),(r_l - 1)) (l \geq 2)$$
$$e_{i,l} d_{i,l} = 1 \ (\text{mod } L_{i,l}),$$

by using {rl}. For the sake of convenience, we denote ni = piqi(= $n_{i,1}$) and ei = ei,1.. Let $n_{min} = \min\{n_i\}i$ and h1 be a hash function to Zn_{min}, where min{ni}i means the minimum integer of {ni}i. In RSA-based multisignature, both operations in $Zn_{i,l}$, * and $\ddot{O}$ are set to Φ (EOR) and Ii's signature sgn_i is just the next input to I_{i+1}'s signature generation.

Signature generation: The first signer I_1 generates a signature on an original message m1: select a minimum number $n_{1,l1}$ such that $n_{1,l1} > h1(m_1||ID_1)$ and compute $sgn_1 = (h1(m_1||ID_1))^{d_1,l_1} \ (\text{mod } n_{1,l1})$. Then send (ID_1, m_1, l_1, sgn_1) as a signature on mi to the next.

A signer Ij receives m1,m2, ...,mi-1 from Ij-1. If j > 2, patch the message $M_{1,2,...,j-1}$ = Patch(m1,m2, ...,mj-1), modify it to $M_{1,2,...,j}$. Then Ij generates a signature on

$$mj = Diff (M_{1,2,...,j-1}, M_{1,2,...,j-1,j}):$$

select a minimum number $n_{j,lj}$ such that $n_{j,lj} > sgn_{j-1}$ Ö h1(mj ||IDj) and compute $T = sgn_{j-1}$ Ö h1(mj ||IDj), and $sgn_j = T^{dj,lj} \ (\text{mod } n_{j,lj})$.

A multisignature on
$M_{1,2,...,i}$ = Patch(m1,m2, ..,mi) by I1, • • •
Ii-1 and Ii is given by (ID_1, l_1, m_1),
(ID_2, l_2, m_2), • • • and (ID_i, l_i, m_i, sgn_i).

Signature verification: The verifier receives (ID_1, l_1, m_1), (ID_2, l_2, m_2), • • • and (ID_i, l_i, m_i, sgn_i) from a signer I_i.
For j = i, i - 1, ..., 2; compute
$T' = (sgn_j)^{ej,lj} \ (\text{mod } n_{j,lj})$,
$sgn_{j-1} = h1(mj \ ||IDj)$ Ö T'
by using Ij 's public key $(n_{j,lj}, e_{j,lj})$.
Let j = j - 1 and repeat step2.
Compute $T' = (sgn_{1)}^{e1,l1} \ (\text{mod } n_{1,l1})$ by using I1's public key $(n_{1,l1}, e_{1,l1})$ and check
$T' = h1(m_1||ID_1)$. In our multisignature scheme, order of signers does not have to be fixed beforehand. Therefore even if all public keys {ni} are set to be the same size, such a case as $n_{i+1} < n_i$ may happen. This is why we need an additional set of {rl}. The number of signers in a series of multisignatures might be limited according as {rl}. However from a practical point of view, it does not seem to cause a serious problem considering the number of signers for a series of multisignatures. Our multisignature based on RSA has the following features:

- The size of multisignature keeps low even if the number of signers increases.
- It is necessary for each signer to have plural pairs of secret and public key.

Further discussion: We discuss how to add the following feature to our multisignature scheme.

Robustness: If the signature verification fails, then prevent such an unauthentic message from damaging a receiver.

We realize robustness by combining our multisignature with an encryption function. So we call it Multiplesigncrypt. Multiplesigncrypt has a feature that a message cannot be recovered if the signature verification fails, in addition to order flexibility[10].

Therefore a multisigncrypt can prevent computer virus mixed into a message from damaging a receiver since unauthentic message can not be recovered.

Multiplesigncrypt scheme: A center publishes two hash functions h1 and h2 and an encryption and the decryption function, $E(K_i, m_i)$ and $D(K_i, C_i)$, in addition to initialization in basic multisignature scheme, where h2 is used for computing a session key K_i for E and D and C_i is a cipher text. Figure 3 and 4 show the signature generation and verification, respectively.

Signature generation: The first signer I_1 computes $sgn_1 = sign(sk_1, h1(m_1||ID_1)) = (r1, s1)$, where sgn_1 is divided into two parts of r1 and s1. r1 is the next input to

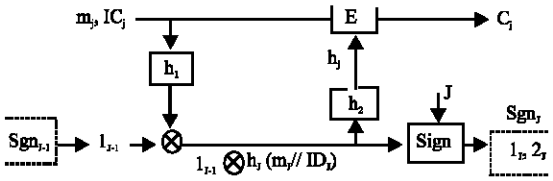

Fig. 3: Ij's signature generation

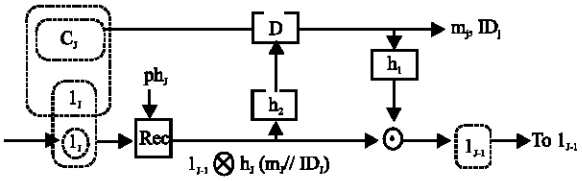

Fig. 4: Ij's signature verifIcation step

I_2's signature generation, which can be recovered by I_2's signature verification.On the other hand,s1 is the rest of the signature which is sent to all signers as it is.
Then, generate a session key K1,

$$K_1 = h2(h1(m_1||ID_1)),$$

and encrypts $m_1||ID_1$ by an encryption function E,

$$C_1 = E(K_1, m_1||ID_1),$$

and sends (ID1, s1, r1, C1) to the next signer I_2.

A signer Ij verifies the signature on m1, · · · ,mj-1 from Ij-1 according to the verification step in the next paragraph and modifies M1,··· ,j-1 = Patch(m1, · · · ,mj-1) to M1,··· ,j. Then Ij generates a signature on the difference

$$mj = Diff(M_{1,\cdots,j-1}, M_{1,\cdots,j-1,j})$$

compute
$sgn_j = Sign(sk_j, r_{j-1} * h1(mj||IDj))$
$= (rj, sj)$,
$Kj = h2(r_{j-1} * h1(mj||IDj))$,
and encrypts mj ||IDj by using the session key Kj ,
$Cj = E(K_j, mj||IDj)$.

A multisignature on
$M_{1,2,\cdots,i}$ = Patch(m1,m2, ..,mi)
by I1, ·· , Ii is given by (ID1, s1,C1),
(ID2, s2, C2), · · ·, (IDi, si, ri, Ci).

Signature verification: The verifier receives (ID1, s1, C1), · · · $(ID_{i-1}, s_{i-1}, r_{i-1}, C_{i-1})$, (ID_i, s_i, r_i, C_i) from the signer Ii.
For j = i, ..., 3, 2: compute
$Tj = Rec(pkj, (sj, rj))$ and $Kj = h2(Tj)$,
and decrypts mj and IDj by
$m'_j||ID'_j = D(K_j, C_j)$.

If $ID'_j = Id_j$ holds, then accept the signature and recover r_{j-1},
$r_{j-1} = Tj ? h1(m'j||ID'j)$.
Set j = j - 1 and repeat step 2
Compute T1 = Rec(pk1, (s1, r1)) and
K1= h2(T1) and decrypt m_1 and ID_1 by
$m'_1||ID'_1 = D(K_1, C_1)$.
If $h1(m'_1||ID'_1) = T1$ holds, then
accept the signature and finally patch all
messages,
$M_{1,\cdots,i} = Patch(m1, \cdots, mi)$.
Thus, we can recover the message if
and only if it is proved to be an
authenticated message.

CONCLUSION

In this study, we have proposed a new multisignature scheme suitable for circulating messages through internet. Our multisignature scheme realizes four features: robustness, order flexibility, multiplesigncrypt and maintaining the multiple signature with the same length as a single signature. Furthermore, we have proved that the security has been improved better than the previous multisignature schemes by employing multiplesigncrypt which prevents unauthenticated message from damaging the receiver.

REFERENCES

1. Noakes-Fry and Kristen, 2003. Digital Signatures: Perspective. August 10, 2000. URL: http://www.securitytechnet.com/resource/rsc-center/gartner/digitalsigs.pdf.
2. Diffie, W. and M.E. Hellman, 1976. Multiuser Cryptographic Techniques. AFIPS Conf. Proc., 45: 109-112.
3. Leung, Karl R.P.H. and C.K. Hui Lucas, 2003. Multiple Signature Handling in Workflow Systems. 2000. URL: http://www.computer.org/proceedings/hicss/0493/04936/04936033.pdf.
4. Rubin and Frank, 2003. A Multiple Signature Protocol for Public-key Cryptosystems. August 6,1997. URL: http://www.contestcen.com/crypt004.htmt.
5. Itakura, K. and K. Nakamura, 1983. A public-key cryptosystem suitable for digital multisignatures. NEC J. Res. Dev., pp: 71.
6. Chafic Maroun, 2003. Rouhana Moussa Digital Signature and Multiple Signature: Different Cases for Different Purposes.

7. Wheatman, S. Victor, Noakes-Fry and Kristen, 2003. Digital and Electronic Signatures: A Quick Look. May 16, 2003. URL: http://www4.gartner.com/2_events/audioconferences/attachments/sd_16may03.ppt
8. Rivest, R., A. Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryp-tosystems, Communications of the ACM, pp: 120-126.
9. Rubin, F., 1995. Message Authentication Using Quadratic Residues. Cryptologia, 19: 397-404.
10. Mitomi and Shirow, 2001. MIYAJI, Atsuko. A General Model of Multisignature Schemes with Message Flexibility, Order Flexibility and Order Verifiability. October 10 2001. http://grampus.jaist.ac.jp:8080/miyaji-lab/member/PaperPS/IEICE01-sita.pdf.