of 15 μs. This suggests therefore that the system could safely be used in most geographical locations of interest. In mountainous regions with very long echo delays, some gain would still be achieved by setting τ to T/2 without significantly degrading the receiver equalisation capability.

The technique has also been validated through testing with more complex Rayleigh fading channel models specified by GSM which are typical of particular locations. For example, a gain of over 6 dB at a bit error rate of 10^-2 has been found for the ‘rural area’ model with τ = T.

Diversity system configuration: This diversity scheme may be implemented both at the BS (base station) and at the MS (mobile station), using different configurations which are briefly reviewed below.

(a) Receiver diversity: This is a direct realisation of the system illustrated in Fig. 1, which could be implemented at either the BS or the MS. For the MS, only a small spacing between the two antennas is required to achieve good fading decorrelation. To comply with the GSM equaliser specification, MS manufacturers might consider several possible configurations, e.g. (i) a small fixed delay of T/2 giving reduced diversity gain but enabling compliance, or (ii) provision of a more complex equaliser designed to cope with up to 20 μm dispersion (i.e., the specified 10 μm plus one bit period). A more sophisticated approach would consist of allowing the delay τ itself to be set adaptively after sensing the channel dispersion. Thus τ could be reduced and eventually nulled (no diversity) under extreme dispersive conditions.

(b) Transmitter diversity: Assuming reciprocity, it follows that diversity in the BS to MS direction could be generated by the BS itself using the arrangement of Fig. 1 for transmission, effectively providing two transmission paths with a relative time delay large enough for the equaliser to separate them at the MS. The resulting gain should be identical to that found for receiver diversity. The main advantage of this mode of operation lies in the fact that the MS would only require one antenna with no extra processing. However, a handset designed according to the GSM specifications might not operate correctly in areas exhibiting large delay spreads, and the BS would therefore need to adapt the delay τ to ensure that excessive dispersion was not generated, or alternatively implement only a small fixed delay T/2 as discussed above for receiver diversity.

Conclusions: A combined space/time diversity technique has been described which does not require any special switching or combining arrangements as well as avoiding the complexity of frequency hopping based schemes. This technique is particularly suited to narrowband TDMA links where the receiver processing is likely to include equalisation, which may be directly used to resolve and combine the diversity paths.

L. B. LOPES
19th May 1989
Department of Electrical & Electronic Engineering
University of Leeds
Leeds LS2 9JT, United Kingdom

References
5. 'Transmission and reception.' CEPT/GSM Recommendation 05/05.

NEW SCHEME FOR DIGITAL MULTISIGNATURES

Indexing terms: Codes, Information theory

When two individual users wish to carry on a secure conversation, they can use the well-known RSA public key cryptosystem in doing so. This cryptosystem provides to these users both data secrecy and digital signature in a very efficient manner. However, in many applications, multiple users need to sign a document. In this letter, we propose a modified scheme, based on the RSA system, which will allow any number of users to sign a document and send it secretly to the receiver. The length of ciphertext remains constant, no matter how great the number of signatories. The trade-off is that the processing times required for generating the multi-signature, and for verifying multisignatures, depend on the number of signatories.

Introduction: In 1976, Diffie and Hellman invented the concept of the public-key distribution cryptosystem. Since then, several public-key systems have been proposed. Among them, the RSA scheme has received the widest attention. For this RSA scheme, the difficulty of breaking the system is based on the difficulty of factoring a large integer into its two large prime factors. This scheme provides both data encryption and digital signature for one-to-one communication with bit ratios of 1:1.

However, for some applications, there are multiple users involved in signing a document. This problem is the so-called ‘multisignature’ problem. Itakura and Nakamura have developed a solution based on an extended RSA scheme. In their method, any user must first join a system by registering with a key generating centre, which centre distributes to that user a set of public keys and a set of secret keys. In addition, a public key which reflects the user’s position in the system must also be generated by the centre at registration time. There are two major drawbacks in their proposed scheme, which make it inappropriate outside its intended realm of application, i.e. within certain types of organisations. First, the key generation process is centralised. For most current networks, where there are millions of users involved, this arrangement is obviously impractical. Second, for most networks, a hierarchical relationship among users either cannot be predetermined or does not exist, rendering this method unsuitable.

Boyd proposed a double signature scheme. He states that this method, however, cannot be extended to apply to the case of more than two signatories, without requiring that all signatories other than the first and last ones be willing to sign a document that they cannot see, i.e. be ‘blind signatories’.

In this letter, we propose a modified RSA scheme which can allow any number of users to sign a document and send it secretly to the receiver. The length of ciphertext remains constant no matter the number of signatories. One major feature in our method is the following: since the signatories need to sign the document consecutively, it becomes necessary to apply a succession of transformations on intermediate signatures using different moduli, which moduli must therefore be ordered; hence, we incorporate the ideas proposed by Rivest et al. and Kohnfelder to provide this ordering of moduli, thus eliminating any need to reblock intermediate signatures. In other words, if the RSA scheme is already implemented on the system, then our scheme allows network users to use that scheme for the additional task of providing multisignatures when needed.

Our proposed method for providing data secrecy and multisignature:

Public/secret keys: Each user has to select randomly two sets of two large primes as his/her secret key, (pA1, qA1) and (pA2, qA2), and then to evaluate the corresponding public keys as nA1 = pA1qA1 and nA2 = pA2qA2, where nA1 < h < nA2, and h is a special system threshold value which is publicly known. Two other sets of keys also need to be calculated. The first set,
(E₄, D₄), is used for providing signatures, and the second set, (E₅, D₅), is used for proving secrecy. These keys must satisfy the following conditions:

\[ E₄D₄ \mod \Phi(n₁) = 1 \]

\[ E₅D₅ \mod \Phi(n₂) = 1 \]

where \( \Phi \) is the Euler totient function. Finally, (E₆, q₁, E₇, q₂, E₈, D₈) needs to be made publicly known and (p₁, q₁, p₂, q₂, E₉, D₉) needs to be kept as a personal secret.

**Encryption for multisignature**: We think that it is reasonable that any document which needs to be signed by multiple users be prepared by one or more of those users whom we call the 'initiator'. The initiator needs to determine all the signatories and to access their public keys. Then the signing order is determined according to their public key values. The signing order for signatories \( U_i \) and \( U_j \) is that if \( U_i \) precedes \( U_j \) whenever \( n_{i-1} < n_{j-1} \).

**Multisignature generation**: Without loss of generality, we will assume that there are \( r \) users \( U_1, U_2, \ldots, U_r \) who need to sign some document, and that their public key values satisfy the inequality \( n_{i-1} > n_{j-1} \). Note that only their first public key elements, \( n_{i-1} \), the ones used for providing signatures, are involved.

**Step 1**: The initiator needs to send this signing order information, and the document, to the first signatory on the list. Therefore, the initiator needs to compute \( C_1 = E_{U_1}(M) \) and send \((U_1, U_2, \ldots, U_r, C_1)\) to \( U_1 \). Note that we use the expression \( E_{U_1}(M) \) to indicate the computation \((M^{n_{U_1}} \mod n_{U_1}) \) done by \( U_1 \).

**Step 2**: \( U_1 \) first deciphers the encrypted message \( C_1 \) as \( M = D_{U_1}(C_1) \). Then if \( U_1 \) agrees to sign the message, he/she will put the signature on this message \( S_1 = D_{U_1}(M) \) and send this signature secretly to the second signatory on the list. Therefore, \( U_1 \) needs to compute \( C_2 = E_{U_2}(S_1) \) and send \((U_1, U_2, \ldots, U_r, C_2)\) to \( U_2 \).

**Step 3**: \( U_2 \) first will decipher the encrypted message \( C_2 \) by computing \( S_2 = D_{U_2}(C_2) \) and storing \( S_1 \). Then he/she will decipher \( S_1 \) again to see the plaintext message \( M = E_{U_1}(S_1) \). Since \( n_{i-1} > n_{j-1} \), the original message \( M \) will be recovered. Then if \( U_2 \) agrees to sign the message, he/she will put his/her signature on the \( S_1 \) value stored previously, as \( S_2 = D_{U_2}(S_1) \) and send this double signature which involves both \( U_1 \) and \( U_2 \) to the third signatory on the list. Therefore, \( U_2 \) needs to compute \( C_3 = E_{U_3}(S_2) \) and send \((U_1, U_2, \ldots, U_r, C_3)\) to \( U_3 \).

**Step (i + 1) (3 ≤ i ≤ r - 1)**: \( U_i \) first will decipher the encrypted message \( C_i \) by computing \( S_{i-1} = D_{U_{i-1}}(C_i) \) and storing \( S_{i-2} \). Then he/she will decipher \( S_{i-1} \) again to see the plaintext message \( M = E_{U_{i-1}}(S_{i-1}) \). Since \( n_{i-1} > n_{j-1} \), the original \( M \) will be recovered. Then if \( U_i \) agrees to sign the message, he/she will put his/her signature on \( S_{i-1} \) as \( S_i = D_{U(i)}(S_{i-1}) \) and send this multisignature \( S_i \) which involves \( U_1, U_2, \ldots, U_i \) to the \((i + 1)\)th signatory on the list. Therefore, \( U_i \) needs to compute \( C_{i+1} = E_{U_{i+1}}(S_i) \) and send \((U_1, U_2, \ldots, U_i, C_{i+1})\) to \( U_{i+1} \).

**Step (i + 1)**: \( U_{i+1} \) first will decipher the encrypted message \( C_i \) by computing \( S_{i+2} = D_{U_{i+2}}(C_{i+1}) \) and storing \( S_{i+1} \). Then he/she will decipher \( S_{i+1} \) again to see the plaintext message \( M = E_{U_{i+1}}(S_{i+1}) \). Then \( U_{i+1} \) agrees to sign the message, he/she will put his/her signature on \( S_{i+1} \) as \( S_i = D_{U_{i+1}}(S_{i+1}) \) and send this multisignature which involves all \( i \) signatories, to the legitimate receiver \( R \). Therefore, \( U_i \) needs to compute \( C_{i+1} = E_R(S_i) \) and send \((U_1, U_2, \ldots, U_i, C_{i+1})\) to \( R \) where \( E_R \) is the public key for data enciphering purposes of receiver \( R \). Note that we use here the second public key of the receiver, since it functions for secrecy, not digital signature. It is to be noted also that \( n_{i+1} > n_{i-1} > n_{i-2} > \cdots > n_{i-1} \).

**Multisignature verification and message decryption**: User \( R \) will decipher the multisignature as \( S_i = D_R(C_{i+1}) \) and store \( S_i \), which involves the secret keys of \( U_1, U_2, \ldots, U_i \), as a proof of their multisignature of the message to be revealed. Then the corresponding signed message can be derived from \( S_i \) by computing \( M = E_{U_i}(E_{U_{i-1}}(E_{U_{i-2}}(\cdots (E_{U_2}(E_{U_1}(S_1)))) \cdots )) \).

**Conclusion**: In this letter we have proposed a cryptographic algorithm which can provide both data secrecy and multisignatures based on the well known RSA cryptosystem. Without requiring any extra investment, our scheme provides this additional ability to network users on any network system which already provides for the use of the RSA cryptosystem.