This typical example shows that the analysis method is very efficient and considerably simpler than other known methods. Although it is presently restricted to stray-insensitive SC networks, it can easily be extended,[8] as will be described in a forthcoming comprehensive publication. There it will be shown that this method can readily be used for general SC networks, including those with unity-gain buffer amplifiers.

A. DĄBROWSKI 4th January 1989

Institute of Electronics & Telecommunication
Technical University of Poznań
ul. Piotrowo 3a, PL-60 965 Poznań, Poland

G. S. MOSCHYTZ

Institute of Signal & Information Processing
Swiss Federal Institute of Technology
ETH Centre, CH-8092 Zurich, Switzerland

References

1 MOSCHYTZ, G. S., and MULAWKA, J. J.: 'Direct analysis of stray-insensitive switched-capacitor networks using signal flow graphs', *IEE Proc. G, Electron. Circuits & Syst.*, 1986, **133**, pp. 145–153
2 BEDROSIAN, S. D., and REFAI, S.: 'A flow graph transformation technique specifically for discrete time networks', *J. Franklin Inst.*, 1983, **315**, pp. 27–36
3 BEDROSIAN, S. D., and REFAI, S.: 'Switched-capacitor networks: analysis and fault diagnosis'. Proc. IEEE int. large scale systems symp., Virginia Beach, Oct. 1982, pp. 316–320
4 CICHOCKI, A., and UNBEHAUEN, R.: 'Simplified analysis of arbitrary switched-capacitor networks', *IEE Proc. G, Electron. Circuits & Syst.*, 1987, **134**, pp. 45–53
5 BON, M., and KONCZYKOWSKA, A.: 'All-symbolic analysis techniques for multiphase switched capacitor networks'. Proc. Europ. conf. on circuit theory and design, The Hague, The Netherlands, Aug. 1981, pp. 655–660
6 MOSCHYTZ, G. S., and BRUGGER, U. W.: 'Signal-flow graph analysis of SC networks', *IEE Proc. G, Electron. Circuits & Syst.*, 1984, **131**, pp. 72–85
7 HOKENEK, B. S., and MOSCHYTZ, G. S.: 'Analysis of multiphase switched-capacitor (m.s.c.) networks using the indefinite admittance matrix (i.a.m.)', *ibid.*, 1980, **127**, pp. 226–241
8 DĄBROWSKI, A.: 'A method for by-inspection derivation of signal flow graphs for biphase switched-capacitor networks'. Report 8703, Swiss Federal Institute of Technology, Zurich, Institute of Signal and Information Processing, Nov. 1987

PUBLIC-KEY ENCRYPTION ALGORITHM INCORPORATING ERROR DETECTION

Indexing terms: Information theory, Codes and coding, Public-key cryptography, Error-detection codes, Quadratic residue, Authentication

Owing to their mathematical properties, quadratic residues have been used successfully in designing a number of cryptographic applications, such as oblivious transfer protocol and coin flipping protocol. In the letter we propose an encryption scheme based on quadratic residue theory. In particular, we incorporate the encrypting procedure and error-detecting code into a complete communication system.

Introduction: In 1976, Diffie and Hellman[1] introduced the concept of public-key cryptography, which provides a proper solution to the problem of key distribution. Since then many implementations of public-key cryptography have been proposed. For example, the Rivest–Shamir–Adelman (RSA) scheme[2] depends on the difficulty of factoring large integers, and Elgamal's scheme[3] depends on the difficulty of computing discrete logarithms.

In this letter we propose a public-key encryption/decryption scheme that incorporates into it an error-detecting code. The public/private keys are based on quadratic residue theory. The oblivious transfer protocol proposed by Blum[4] is one application based on the same theory. The security of our system is based on the difficulty of factoring large integers, as in the RSA scheme.

Diffie and Hellman[5] observed that if an error is propagated by an encryption/decryption algorithm, then applying error-detecting codes before encryption and after decryption (as shown in Fig. 1) provides a way to achieve message authenticity (to the extent of detecting active tampering). The reason for this is that any altering of the ciphertext will be detected by the error-detecting device.

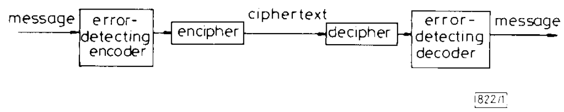

Fig. 1

Since we use the block cipher for encryption, the error does propagate. One special feature of our proposed encryption scheme is that we incorporate error detection within our decryption technique. The result is that our scheme provides not only data secrecy but also, as a bonus, error detection and message authenticity.

Quadratic residues: We now summarise those facts about quadratic residue theory (cf. Denning's summary (Reference 6, pp. 111–117)) needed to understand our encryption algorithm presented later. Those facts are as follows:

(1) *Definition:* A number a is a quadratic residue modulo n iff (a) a and n are relatively prime (gcd $(a, n) = 1$), and (b) the equation $x^2 \bmod n = a \bmod n$ has a solution.

(2) *Notation:* We define QR_n to be the set of all integers between 1 and $n - 1$ that are quadratic residues modulo n, and NQR_n those that are not, called quadratic nonresidues.

(3) Since $(n - a)^2 \bmod n = a^2 \bmod n$ for $a = 1, 2, \ldots, \lceil((n-1)/2)\rceil$, QR_n can be found by evaluating $x^2 \bmod n$ for only $x = 1, 2, \ldots, \lceil((n-1)/2)\rceil$. If n is a prime, these $\lceil((n-1)/2)\rceil$ values will all be distinct. If $n > 2$ and prime, then QR_n contains exactly half of these $n - 1$ elements.

(4) For prime $n = p > 2$, and $0 < a < p$, $a^{(p-1)/2} \bmod p = 1$ or $p - 1$, depending on whether $a \in QR_n$ or $a \in NQR_n$, respectively.

(5) If $n = p^*q$, where p and q are large primes, then $a \in QR_n$ iff $(a \in QR_p)$ and $(a \in QR_q)$. If $a \in QR_n$, then a has exactly 2 or 4 square roots. Since square roots come in additive inverse pairs, half of these roots are even and half odd. We call the even root(s) the primary square root(s) of a.

(6) In this letter we make the further assumption that, for $n = p^*q$, p and q are large primes such that $(p + 1)$ and $(q + 1)$ are each evenly divisible by 4. This assumption considerably simplifies the calculations required for finding square roots (see Reference 6, p. 116).

The reason why quadratic residues, and primary square roots of quadratic residues, play a significant role in our secrecy scheme is threefold:

(i) Quadratic residues modulo n within the integer interval $[1, n - 1]$ fall randomly in that interval.[7]

(ii) Primary square roots of a quadratic residue modulo n are independent of each other.

(iii) Given a quadratic residue a modulo n, where $n = p^*q$, with p and q large prime numbers, it is computationally infeasible to calculate the square roots of a without knowing p and q.

Encryption scheme: The following cryptosystem, based on the theory of quadratic residues, provides secrecy as well as authentication of the message itself (freedom from active tampering). It also incorporates error detection within the decryption algorithm.

Each user U of the system selects two very large (of the order of 350 bits in length) prime numbers p_U and q_U, with the added property that $p_U + 1$ and $q_U + 1$ are each divisible by 4, and then calculates $n_U = p_U * q_U$. The user then makes n_U public while keeping p_U and q_U secret.

Any message M to be sent is divided into a sequence of message blocks of uniform length L, $(M_1, M_2, \ldots, M_k)$, each of which can be considered as a large integer in binary form. Individual blocks are then encrypted. We henceforth refer to individual message blocks as M.

We need a method for encrypting these individual message blocks M. Two earlier papers in particular[5,8] directed our thinking about this problem. In Reference 5 Diffie and Hellman recommended that error-detection encoding and decoding precede and follow, respectively, the encipherment and decipherment of a message, as shown in Fig. 1 (see Reference 6, p. 137).

Our method instead incorporates the encoding and decoding into a system, and in particular requires the use of error-detection decoding in the actual decipherment of the message.

In Reference 8 Koyama presented a method for encrypting messages, which preserves secrecy of the message and requires that the receiver use the quadratic formula to find the roots of a quadratic equation. He appended redundant information to the message, so that this redundant information can be used to select the one of the four solutions to the quadratic equation which includes the plaintext. Instead of using redundant information, we use parity bits already necessary for error-detection encoding and decoding, and one special parity bit of our own, to enable us to select the one square root of the ciphertext that contains the plaintext.

The algorithm for enciphering a block of plaintext M is as follows:

(1) Find the public value n_B of the receiver B, to whom the message is to be sent.

(2) Break the message into a sequence of blocks, all of the same length L, such that the corresponding parity bit string required for a block of length L, being of length t (to produce an $(L + t, L)$ linear block code for error-detection purposes[9]) is such that $(L + (t + 1)) < \log_2 n_B$.

(3) For each message block M, (a) append to M a string P of parity bits and the single bit '0', as follows to produce $M' = P$ @ M @ '0' (we use the symbol '@' for concatenation of strings). M', considered as an integer, is less than n_B, because of the restriction imposed in (2) above on L; (b) calculate the corresponding ciphertext C as $C = (M')^2 \bmod n_B$, and transmit C to user B.

Note that, for this scheme, user A needs basically perform only one simple operation other than those already required for communication protocol, including error detection.

User B then deciphers a message block M from a ciphertext C, as follows:

(1) Find the four (or two) square roots of C.

(2) Eliminate the odd square roots (root), thus obtaining the primary root(s).

(3) If necessary, select M' from the remaining two primary roots.

(4) Remove '0' and P from M' to obtain the message block M.

Steps (2) and (4) are automatic. Steps (1) and (3) require calculations, which will now be described. Step (3) uses decoding, which would be required in any case. Hence, step (1) is where the time complexity of this deciphering algorithm lies.

More specifically, user B, knowing the two factors p_B and q_B of n_B can calculate the four square roots of C in polynomial time. The method involves finding two inverses, one modulo p_B and one modulo q_B, and then using the Chinese remainder theorem twice. These square roots will be of the form r_1, $n_B - r_1$, r_2 and $n_B - r_2$. Since n_B is odd, two of these will be odd and two even, so we reject the odd ones, and call the remaining even ones er_1 and er_2.

The task now is to determine which of er_1 and er_2 is M' (if there were only two roots, instead of four, we would now have found M'). Suppose that $er_1 = P_1$ @ M_1 @ '0' and $er_2 = P_2$ @ M_2 @ '0'. Replace er_1 and er_2 by s_1 and s_2, respectively, where $s_1 = P_1$ @ M_1 and $s_2 = P_2$ @ M_2. Apply the decoding parity-check matrix to s_1 and s_2 to detect any errors in transmission. Whichever one is actually P @ M will pass, while the other will fail with high probability, since it is highly unlikely that a random string s would have its first t bits to be the correct parity bits for the remaining L bits. This probability of failure is $1 - (1/2^t)$. This can be made as close to 1 as is desired by choosing the length of the parity bit string to be large. Note that the decoding of s_1 and s_2 not only checks for errors in transmission, but at the same time enables user B to select the correct square root M'. Hence we have merged the two operations of decoding and decryption (shown in Fig. 1) into a single operation.

If the ciphertext C had no error in transmission, M would be recovered with high probability. If there were an error in transmission, then both s_1 and s_2 would be rejected with high probability, since this error is propagated by the deciphering process, and user B would request user A to retransmit the message. Also, if an enemy were to attempt to actively tamper with the ciphertext C, then again the tampering error would be propagated in the received ciphertext C', leading to two values s_1' and s_2', which would both be rejected with high probability. So, in both the case of active tampering, and of 'natural' tampering, the resulting ciphertext would be rejected, and user A would be requested to retransmit the message. User B would not know which case produced the error, but would detect it in either case with high probability.

Hence, this scheme provides secrecy (no user besides user B knows p_B and q_B, which are necessary to calculate the four square roots of C in nonexponential time; hence no other user could recover M) and as a bonus, message authentication (any other user who attempted to actively tamper with the message would cause user B to request retransmission of the message by user A). It does not authenticate the sender of the message, however, and hence does not either provide a signature for the message.

Conclusion: In this letter we have presented a new cryptographic scheme. This encryption scheme assures secrecy of a message and message authenticity, while incorporating coding and cryptography into a single communication system. This scheme is based on the exponential time complexity of factoring a large integer into two large, prime factors, and on quadratic residue theory in number theory. The overall complexity is equivalent to the familiar Rivest–Shamir–Adelman (RSA) scheme.[2] In the RSA scheme, the computational load is equally divided between the sender and receiver. In our scheme, the sender has a very light load, while the receiver bears a heavy computational load.

L. HARN
T. KIESLER

14th November 1988

Computer Science Program
University of Missouri–Kansas City
Kansas City, MO 64110, USA

References

1 DIFFIE, W., and HELLMAN, M. E.: 'New directions in cryptography', *IEEE Trans.*, 1976, **IT-22**, pp. 644–654
2 RIVEST, R. L., SHAMIR, A., and ADELMAN, L.: 'A method for obtaining digital signatures and public-key cryptosystem', *Commun. ACM*, 1978, **21**, pp. 120–126
3 ELGAMAL, T.: 'A public key cryptosystem and a signature scheme based on discrete logarithms', *IEEE Trans.*, 1985, **IT-31**, pp. 469–472
4 BLUM, M.: 'Three applications of the oblivious transfer: 1. Coin flipping by telephone. 2. How to exchange secrets. 3. How to send certified mail'. Dept. EECS, Univ. of California, Berkeley, CA, 1981
5 DIFFIE, W., and HELLMAN, M. E.: 'Privacy and authentication: an introduction to cryptography', *Proc. IEEE*, 1979, **67**, pp. 397–427
6 DENNING, D. E. R.: 'Cryptography and data security' (Addison-Wesley, Reading, MA, 1982)
7 KNUTH, D. E.: 'The art of computer programming, vol. 2: seminumerical algorithms' (Addison-Wesley, Reading, MA, 1969)
8 KOYAMA, K.: 'A master key for the Rabin's public-key cryptosystem', *Trans. Inst. Electron. & Commun. Eng. Jpn.*, 1983, **J66-D**, pp. 1362–1369
9 LIN, S., and COSTELLO, D. J.: 'Error control coding: fundamentals and applications' (Prentice-Hall, Englewood Cliffs, NJ, 1983)