[10] Carhoun, D. O., Johnson, B. L., Meehan, S. J., "Design and Hardware Implementation of a Versatile Transform Decoder for Reed-Solomon Codes," presented at the 1981 IEEE International Symposium on Information Theory, Santa Monica, CA, 1981.

[11] Carhoun, D. O., Johnson, B. L., Meehan, S. J., "VLSI Architectural Design for a Reed-Solomon Transform Decoder," presented at the 1981 IEEE International Symposium on Circuits and Systems, Chicago, IL: 1981.

[12] Carhoun, D. O., Johnson, B. L., Meehan, S. J., "Transform Decoding of Reed-Solomon Codes Volume I: Algorithm and Signal Processing Structure," ESD-TR-82-403, Vol. I, November 1982, ADA123953.

[13] Johnson, B. L., Bequillard, A. L., Meehan, S. J., "Transform Decoding of Reed-Solomon Codes Volume II: Logical Design and Implementation," ESD-TR-82-403, Vol. II, November 1982, ADA123977.

[14] Reed, I. S., Scholtz, R. A., Truong, T. K., and Welch, L. R., "The Fast Decoding of Reed-Solomon Codes Using Fermat Theoretic Transforms and Continued Fractions," IEEE Transactions on Information Theory, IT-24, pp. 100-106, 1978.

[15] Mandelbaum, D., "Construction of Error Correcting Codes by Interpolation," IEEE Transaction on Information Theory, Vol. I, pp. 27-35, January 1979.

[16] MacWilliams, F. S., Sloane, N. J. A., The Theory of Error Correcting Codes, North Holland, New York, 1977.

[17] Ferguson, T. J., Johnson, B. L., Carhoun, D. O., "Implementation of a Transform Decoder for Reed-Solomon and Alternant Codes" presented at the 1983 IEEE International Conference on Communications, Boston, MA, June 1982.

[18] Paczan, M. W., and Johnson, B. L., "LSI Design of a Programmable Polynomial Residue Calculator for a Reed-Solomon Transform Decoder," presented at the 1982 IEEE Custom Integrated Circuits Conference, Rochester, NY: 1982.

PROCESSING TECHNIQUES IN PUBLIC KEY CRYPTOSYSTEMS

Rod Goodman

Department of Electronic Engineering
University of Hull, Hull HU6 7RX , U.K.

Abstract

The increasing use of cryptographic techniques in business and commercial data communications systems will only come about if cheap and fast hardware LSI devices can be designed to implement the algorithms. This has already happened with the DES but in the case of public key cryptosystems the process is only at the development stage. This is due to this nature of the algorithms and to the fact that the algorithms are themselves under suspicion. The paper examines public key cryptosystems and their modifications from an implementation point of view.

1. Introduction

In modern cryptography, the security of a transmission rests in the secrecy of a "key" rather than in keeping the algorithm used by the cipher machine a secret. In a conventional (or symmetric) cryptosystem (CC) such as the DES (ref.1), the algorithm is in fact an international standard. In such a system the cipher machine is "primed" with a secret key which tells the machine which transformation to apply to the plaintext, out of the many possible transformations within the algorithm, to turn it into the ciphertext. The receiving cipher machine uses the same (or a directly related) key to effect the inverse transformation from ciphertext to plaintext. An enemy with an identical machine can only try to find or steal the key, given that the cryptosystem has been well designed in that it is computationally and statistically infeasible to deduce the key even with known plaintext ciphertext pairs. A fundamental limitation of this system is therefore that the users must have previously securely set up a common key.

This "key distribution problem" is severe in a networking or electronic mail environment. If a packet or local area network has n users, any pair of whom may wish to communicate, the number of potential keys rises as n squared. There is a further limitation of the CC when authentication is considered. An enciphered order may contain a weak authenticator such as an order number and date, but because the receiver has the ability to create a ciphertext from any messagetext desired, disputes between the two parties cannot be resolved by a judge.

In 1976 Diffie and Hellman (ref.2) and independently Merkle (ref.3) proposed a new cryptographic scheme called a Public-Key-Cryptosystem (PKC) that is essentially asymmetric (ref.4). The elegance of the method stems from the use of different keys for encryption and decryption (EK and DK), and that it is infeasible to derive one from the other. Thus if network users generate key pairs and make their encryption keys public in a secure (say printed) directory the need to distribute keys does not arise. In order to send a message M to a user, we generate the ciphertext C = E(M) by encryption with his (public) encryption key. The user keeps his decryption key secret so that only he can invert the procedure to give M = D(C) = D(E(M)). An enemy is faced with deriving DK from EK, which we have said is "hard". Note that in this scenario prior authentication is assumed, i.e. the very fact that our encryption key is public enables any user to send us messages. We must be sure that the person sending the message is who she says she is.

PKC schemes also permit us to devise authentication procedures such as signing a contract which we cannot later deny, showing that a message has not been tampered with, and establishing identity beyond doubt. In order to achieve this we require the additional property that for all (or nearly all) cryptograms E(D(C)) = C . That is, "decryption" of a message makes sense because most messages are also cryptograms. We can sign a document as follows. User A sends an message M to user B, encrypted with B's public key for security. A then forms a pre-signature which is some function of the plaintext M and is also a valid cryptogram in A's encryption algorithm (ref.5). A decrypts this using his secret key to form the signature S = D(C) which only he can do. This is then sent to B (via encryption for secrecy if neccessary). B uses A's public key to form C = E(D(C)). B operates on the already received message M to form the pre-signature which she then compares to C. If they match then B is sure that M came from A. Furthermore, as no one other than A (including B) could have produced S, A cannot later deny that he signed M. Also B cannot alter M or S without destroying the correspondence C = E(S). There are situations in which keeping the message secret is not desirable in the authentication process. Consider that 'WE' have an ambassador in

an alien country who has to use 'THEIR' PTT to send 'US' messages. 'THEY' will not allow him to send coded messages which 'THEY' cannot read for fear of espionage. 'WE' must be sure that 'THEY' are not tampering with the ambassador's messages. With asymmetric encryption both parties can be satisfied. 'WE' give 'THEM' the decryption key but keep the encryption key secret. The ambassador gives 'THEM' a message M and the resultant cryptogram C. 'THEY' decrypt C and check that D(C) = M. 'WE' receive M and C and check that the message is authentic if E(M) = C.

We can concieve of a PKC system working in conjunction with a CC, particularly in an electronic mail environment. The PKC is used to securely distribute session keys and to authenticate users. A disadvantage of the PKC is that it is essentially one-to-one. That is, we set up a two user secure channel. If we wish to send a broadcast to several users we need to encrypt the same message several times. This redundancy is particularly severe in a packet switched network where group addressing is usually built in. Several authors have attacked this problem (ref.36,40,41,42).

The secrecy of the PKC resides in the one-wayness of the operations involved. The investigation of suitable one way functions has been the subject of intense research since Diffie and Hellman's paper.

2. Trapdoor One-Way Functions

A function y = F(x) is said to be one-way if 1) there is a one-to-one relation between x and y, 2) given x it is 'easy' to compute y, and 3) given y it is 'hard' to compute x. Furthermore in a trapdoor one-way-function it is easy to compute x from y given some secret side information. Diffie and Hellman (ref.2 and 6) describe a key distribution system based on the one-wayness of the discrete exponential and logarithm functions. If p is a prime and a is a primitive element, then for x and y in the range 0 to p-1

$$y = a^{x} \bmod p \qquad \text{and} \qquad x = \log_{a} y \quad \text{over } GF(p)$$

It is easy to compute y given x (ref.7) in about 2log p (base 2) multiplications (ref.8). For example (ref.9) noting that 35 = 10011 in binary, we have

$$a^{35} = a^{32} \cdot a^{2} \cdot a^{1} = ((((a)^2)^2)^2)^2 \cdot a^2 \cdot a$$

which requires seven multiplications. Even if the x's are several hundred bits long it is still easy to evaluate the exponential as

given in (refs.5 and 10). Evaluation of the logarithm is conjectured to be much more difficult requiring of the order of root p steps (ref.8). Tighter bounds are known (refs.11,12,13) but for numbers of the order of 500 bits, it is still computationally infeasible to find the log.

The distribution method works as follows. Given

$$U = a^{u} \bmod p\ ,\quad I = a^{i} \bmod p\ ,\quad K = a^{ui} = a^{iu} \bmod p$$

'You' think up a random number u and tell me U. 'I' think up a random number i and tell 'you' I. 'You' raise I to the power u, 'I' raise U to the power i and we have both calculated the key K. 'They' only know U and I and our one-way function. To find K 'they' need to find either u or i and 'they' are up against a one-way function.

The generation of secure PKC depends on the finding suitable one-way functions with hidden trapdoor information to make the inversion feasible. Such schemes have been proposed by Merkle-Hellman (ref.5), Rivest et al. (ref.14), McEliece (ref.10), Lu-Lee (ref.15), Kravitz and Reed (ref.16),MITRE (ref.17), Gordon (ref.18), etc. These schemes have been subjected to intense scrutiny and some have fallen as a result. Indeed it may prove impossible to devise workable PKC's given the rate at which new 'holes' in the techniques are found, and the rate at which new modifications are proposed to overcome some of the disadvantages. Notwithstanding this, let us now consider the two most popular systems.

3. The Merkle-Hellman Trapdoor-Knapsack PKC

The knapsack problem is a combinatorial problem in which one is given a vector a of n integers (the weight of each possible object in the knapsack) and an integer S which is the sum of a subset of the a's (the actual weight of the knapsack). The problem is to solve for the subset, that is the binary vector x corresponding to S = a * x (ie find which objects are in the knapsack). The general knapsack problem is one in which the coefficients of x are integers instead of 0 or 1, and this problem is known to be NP-complete and therefore 'hard'. However, some knapsacks are easy to solve. For example if a = (1,2,4,8,16...) ie the powers of 2, then x is the binary representation of S. Merkle and Hellman (ref.5) use a knapsack vector a' which is superincreasing. That is, each integer is strictly greater than the sum of all previous integers. For example a' = (171,196,457,1191,2410). Given S' = 3797 we can easily compute S' = 2410+1191+196 ie x = 11010. This 'easy' knapsack is then disguised by k iterations of modular multiplication to produce a trapdoor

knapsack vector a that is 'hard'.Thus

$$a_k = (((\, a'_1 * w_1 \,) \bmod m_1 \,) * \ldots.\,) * w_k \bmod m_k$$

where w_j is invertable modulo m_j, that is $\gcd(w_j, m_j) = 1$.

if the vector a is made public then anyone wishing to transmit a message x would calculate the hard knapsack S = x * a , which the recipient would transform into the 'easy' knapsack S' = x * a' using the secret m and w, Thus:

$$S' = (((\, S * w_k^{-1} \,) \bmod m_k \,) * \ldots.\,) * w_1^{-1} \bmod m_1$$

Using the example from ref.5, chose m=8443 and w=2550, then w^{-1} =3950 by Euclid's algorithm (ref.8). The published knapsack is now a = (5457, 1663,216,6013,7439). Given S = 1663+6013+7439 =15115, we compute S' = 3950.15115 mod 8443 = 3797 as before. The scheme is attractive because encryption is fast, requiring only addition, and also fast decryption schemes have been proposed (ref.28). The original Merkle Hellman scheme proposed n=100 knapsack vectors and a 202 bit modulus thus making the a 202 bit pseudorandom numbers of length 202 bits. The sum S requires a 209 bit representation giving an intrinsic 2.09 data expansion from x to S with k=1 iterations, and a public key size of 20kbits. Attacks on the system have however forced these parameters to be revised upwards. In particular Shamir (ref.18) shows that two or more iterations are definitely needed, and this causes the data expansion to increase by seven bits at each iteration. At present the whole security of the superincreasing trapdoor is in question (refs.20,21, 22,23,24). In particular Desmet et al. (ref.22) find that iterative transformations do not guarantee higher security, and that infinitely many superincreasing decryption keys exist as soon as one exists. Shamir (ref.21) has discovered a technique which will solve a given knapsack with a probability of success that is directly proportional to the density of the subset sums S, where a dense knapsack is one in which nearly all integers in the interval between 1 and the sum of all the knapsack integers are valid subset sums. This makes dense knapsacks unsafe for cryptographic use. McAuley and Goodman (ref.25) have proposed a new trapdoor in a knapsack PKC that is not based on a superincreasing sequence in order to defeat these attacks. However, new results (ref.26) seem to indicate that most cryptographic knapsacks can be solved in polynomial time, even if they are not based on superincreasing sequences. These results further question whether all useful knapsacks can be cracked, and wheather useful ones can be generated and tested.

The inherent expansion in the trapdoor knapsack means that these systems are not well suited to providing public key authentication, because only a small fraction of all possible message words of a typical length lead to a binary solution of the knapsack. Schobi and Massey (ref.27) have proposed a nonbinary solution that overcomes this.

4. The Rivest-Shamir-Adelman (RSA) scheme

The RSA scheme (ref.14) is based on the fact that it is much easier to generate large primes and multiply them together than it is to factor the result. The key generator chooses two large primes p and q which are a few hundred bits long. if n=pq then Euler's function is (p-1)(q-1) that is, the number of integers between 1 and n which have no common factor with n. We then choose a number e relatively prime to (p-1)(q-1) and use Euclids algorithm to find the 'inverse' d via the expression e.d =1 mod (p-1)(q-1). The public key is (n,e) and our secret trapdoor information is d and the factorisation of n. The message text is represented as an integer from 0 to n-1 and the enciphering and deciphering procedures are the modular exponentiations :

$$C = M^{e} \bmod n \qquad M = C^{d} \bmod n$$

We have seen previously that even if the e and d are large the number of multiplications required in the exponentiation is managable, whilst the enemy has a task as difficult as factoring n. For example (ref.6): choose p = 5 and q = 11. Then n = 55 and (p-1)(q-1) = 40. If e = 7 then d = 23 as 7.23 = 1 mod 40. Choosing a message M = 2 :

$$C = 2^{7} \bmod 55 = 2^{1}.2^{2}.2^{4} \bmod 55 = 18$$

$$M = 18^{23} \bmod 55 = 18^{1}.18^{2}.18^{4}.18^{16} \bmod 55$$

$$= 18.49.36.26 \bmod 55$$

$$= 2$$

The RSA method has also been subjected to attacks but has withstood these much better than than the trapdoor knapsack scheme. (refs 20, 29-34). Furthermore the RSA scheme gives digital signatures directly as there is no expansion of the message text. In addition, an elegant probabalistic test (ref.14) gives us a means of generating large primes efficiently, thus makeing the RSA algorithm self-contained and secure.

5. Implementation

Cryptographic algorithms are ideally suited to VLSI implementation because of their computation-intensive nature. In addition, there have been new developments in tamper-proof chips for software protection (ref.35) and these permit the possibility of secure generation of keys, without user intervention. The main implementation of cryptographic systems so far has been the production of DES chips. These are available from several manufacturers in both single chip and chip-set form. For example the Advanced Micro Devices AMZ8068 which gives throughput rates of over 1 Mbyte per second.

The integration of PKC systems is still at the development stage. There are several reasons for this. Firstly the PKC algorithms require more computation than the DES and thus imply lower data throughput rates, but more importantly he algorithms are still under development and their security is still in question. Given this fact it is not surprising that all implementations have been directed towards the RSA scheme or hybrid DES-RSA schemes where the RSA is used for key distribution and the DES for fast encryption.

Rivest (ref.37) has reported a single-chip implementation of the RSA algorithm. The design is essentially a big-number ALU, that can operate on 512 bit numbers and hence perform all calculations needed by the RSA. The chip implements the operations of addition, subtraction, multiplication, division, remainder, and modular exponentiation. In addition other useful functions such as generation of large primes are performed. The 512 bit ALU is organised in a bit-slice manner with 8 general purpose registers, up-down shifter logic, and multiplier (carry-save) logic. The ALU is only capable of performing the operations A.B+C, shift-left, shift-right, test least significant bit. All the higher functions are implemented by the microprogram stored in the internal PLA. With a feature size of 2 microns the chip is large measuring 5.5 mm by 8 mm, and the 4MHz gives an encryption rate of 1200 bits per second. The complexity of this chip however raises doubts as to the yields obtainable.

The essential operation in the RSA algorithm is that of modular exponentiation. This reduces to modular multiplication at its simplest level. Simmons and Tavares (ref.38) are working on a modular multiplier in NMOS technology using a 6 micron feature size. The design is again bit slice orientated and the two steps of multiplication and then modulo reduction are performed by the same device. That the two operations are essentially the same can be seen as follows. The process of multiplication can be seen as that of conditionally adding together shifted versions of the multiplicand. Thus for each 1 in the multiplier an intermediate

product is formed by adding in a version of the multiplicand that has been shifted the same number of places as the 1 in the multiplier. Modulo reduction can be considered as conditional subtraction of the modulus from the product. The modulus is first shifted left until its MSB is aligned with the MSB of the product. If the shifted modulus is less than the present result, we subtract it to form a new result, and then shift the modulus one bit right. If the shifted modulus is greater we do not subtract but just shift. This repeats until the result is less than the shifted modulus. The implicit comparison operation is complex but can fortunately be eliminated as follows. First, a sign bit is required in the result which is initially presumed positive i.e. s=1. Then starting at the MSB end as before the subtractions are replaced by the addition of one of two values: the modulus or its two's complement. Both values are shifted and the sign of the present result determines which value is to be added. If s=1 ie the result is negative we add the modulus, if s=0 add the two's complement. When the LSB's line up one further addition of the modulus is needed if the result is negative, to ensure a positive final result. Thus a single unit consisting of an adder with inputs that may be shifted can perform both multiplication and modulo reduction. This requires that the input to the adder can be selected from one of four values: zero, the multiplier, the modulus, or the two's complement of the modulus. In ref.38 the authors hope that chips will be ready during 1983. They estimate a total multiply modulo time of 250 microseconds for 128 bit inputs. The design appears attractive, particularly with the incorporation of parallel pipelining of the inputs and outputs. That is, as one set of data is being processed by the arithmetic unit, the result of the previous set is being output and the next data set is being input. The major limitation of the device is its 128 bit maximum wordlength, and its slow throughput of about 4Kbits.

McAuley and Parker (ref.39) have been working on an Advanced Cipher Processor (ACP). The device has a mask allocation number MA743 and is being fabricated by the GEC research laboratories at the Hirst Research Centre, Wembley, England. The device essentially performs modular exponentiation of 512 bit numbers, and is to be fabricated in bulk CMOS technology using 2.5 micron feature size. A data throughput rate of 50 Kbits per second is hoped for.

The VLSI architecture group at Hirst have an active systolic array program and the design of the cipher processor reflects this. The systolic 'data pumping' approach is particularly suitable for high performance computing VLSI structures. In general a systolic array is a one or two dimensional array of identical functional modules, typically simple digital circuits, arranged in a regular fashion. Each module is connected only to its nearest neighbours

for the purposes of data transfer. Each module utilises common control and timing so that all the modules perform the same function simultaneously but on different data items. The data streams move at constant velocity over fixed paths and interact whenever they meet. Multiple use is made of each data item which results in high computational throughput without the need for high bandwidth memory links. The precise function of the cell depends on the problem to be solved.The advantages of the systolic approach include short interconnects thus giving high speed transfer with low power drivers and small chip area, easy-to-scale architecture,small system control overhead, and minimal data transfers to and from memory (data is input once, used, and discarded). Note that this also allows parallel pipelining. The design of the chip builds on previous work at Hirst on an inner-product step cell. This is a circuit for computing the function C' = A.B + C and propagating delayed versions of A and B to neighbouring cells. This work indicated that although bit-parallel arithmetic is faster than bit-serial, the latter approach gives a greater functional throughput per unit chip area. For these reasons, and because of the high density caused by the 512 bit integers aimed at on this single chip, serial arithmetic is used.

The ACP communicates with the host microcomputer via an eight bit bidirectional data bus, and a number of control pins. DMA transfers are supported. Internally the device consists of the control unit, the modular exponential unit, four 512 bit registers which hold the exponent, modulus, inverse and output. Communication with the host is via a 64 byte I/O stack, a control register and a status register. The internal and external operations are essentially asynchronous with communication through the status register, so that for example input data can be loaded to the stack whilst the modular exponential unit is operating. The heart of the unit is the modular exponentiator which consists of a 512 bit serial parallel multiplier which performs the exponentiation and a similar 512 bit divider which performs the modular reduction. The multiplier feeds the divider and vice-versa so that data is only input once and thus input-output pipelining is possible. The data flow is thus circular. To operate the unit the keys are first loaded into the appropriate registers via the I/O stack, that is the modulus, exponent and after a precomputation, the inverse register. The input data is then supplied to the exponentiator which will circulate until the output is ready and in the output register. During this period the I/O stack can be loaded with new data if required. If the result is ready and the stack is still busy, with new data, then the output waits until the new data is claimed by the exponentiator before transferring itself to the I/O stack. This parallel pipelining contributes greatly to the overall speed of the device.

6. References

1 . "The AmZ8068 Data ciphering Processor", Product Description, Advanced Micro Devices, Sept 1980.
2 . W.Diffie and M.E.Hellman, "New directions in cryptography", IEEE Trans. Inf. Th.,vol IT-22, Nov 1976.
3 . R.C.Merkle, "Secure communication over an insecure channel", Common. Ass. Comput. Mach.,vol 21, Apr 1978.
4 . G.J.Simmons, "Cryptology: The mathematics of secure communication", The Math. Intell., vol 1, no 4, Jan 1979.
5 . R.C.Merkle and M.E.Hellman, "Hiding information and signatures in trapdoor knapsacks", IEEE Trans. Inf. Th., vol IT-24, Sept 1978.
6 . M.E.Hellman, "An overview of public key cryptography", IEEE Comm. Soc. Mag., Nov 1978.
7 . A.G.Konheim, Cryptography: A Primer", John Wiley, 1981.
8 . D.E.Knuth, The Art of Computer Programming, Vol 2, Seminumerical Algorithms, Reading, MA: Addison-Wesley, 1969.
9 . J.A.Gordon, "Recent trends in cryptology", Electronics and Power, vol 26, no 2, Feb 1980.
10. R.J.McEliece, "A public key system based on algebraic coding theory", JPL DSN Progress Rep., 1978.
11. S.C.Pohlig and M.E.Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", IEEE Trans. Inf. Th., vol IT 24, no 1, Jan 1978.
12. L.Adelman, "A subexponential algorithm for the discrete logarithm problem with applications to cryptography, Dept. of Math., MIT.
13. A.Shamir and R.Schroepple, "A TS2 = 02n time/space tradeoff for certain NP-complete problems", SIAM J. Comput, vol 10, 1981.
14. R.Rivest, A.Shamir, L.Adelman, "A method for obtaining digital signatures and public key cryptosystems", Comm. ACM, vol 21, 1978.
15. S.C.Lu and L.N.Lee, "A simple and effective public key cryptosystem", COMSAT Tech. Rev. 9, 1979.
16. D.W.Kravitz and I.S.Reed, "Extension of RSA cryptostructure : a Galois approach", Elect. Lett., 18(6), 1982.
17. B.P.Schanning, "Applying public key distribution to local area networks", Workshop on electronic privacy and authentication, The Hatfield Polytechnic, July 1982.
18. J.A.Gordon, "Public key cryptosystems and related topics", Proc. IEE Conf. on data transmission codes, London, Nov 1980.
19. A.Shamir and R.E.Zipple, "On the security of the Merkle-Hellman cryptographic scheme", IEEE Trans. Inf. Th., vol IT 26, no 3,May 1980.
20. T.Herlestam, "Critical remarks on some public key cryptosystems", BIT, vol 18, 1978.

21. A.Shamir, "Cryptocomplexity of knapsack systems", Symposium on the theory of complexity, Atlanta, Georgia, April 1979.
22. Y.Desmet, J.Vandewalle, R.Govaerts, "A critica analysis of the security of knapsack public key algorithms", IEEE Int. Symp. on Inf. Th., Les Arcs, France, Jun 1982.
23. I.Ingemarsson, "A new algorithm for the solution of the knapsack problem", IEEE Int. Symp. on Inf. Th., Les Arcs, France, Jun 1982.
24. A.Shamir, "New results on public key cryptosystems", to appear.
25. A.J.McAuley and R.M.F.Goodman, "Modifications to the trapdoor- knapsack public key cryptosystem", IEEE Int. Symp. on Inf. Th., St. Jovite, Canada, 1983
26. Lagarias and Odlyzko, "Solving low density subset sum problems", Bell Sys. Tech. J., 1983.
27. P.Schobi and J.L.Massey, " Fast Authentication in a trapdoor-knapsack public key system", IEEE Int. Symp. on Inf. Th., Les Arcs, France, Jun 1982.
28. P.S.Henry, "Fast decryption algorithm for the knapsack cryptographic problem", Bell Sys. Tech. J., vol 60, May-Jun 1981.
29. R.R.Rivest, "Critical remarks on critical remarks on some public key cryptosystems by T.Herlestam", BIT, vol 19,1979.
30. B.Blakley and G.R.Blakley, "Security of number theoretic public key cryptosystems against random attack", Cryptologia, vol 3, nos 1 and 2, 1979.
31. G.Simmons and M.Norris, "Preliminary comments on the MIT public key cryptosystem", Cryptologia, Oct 1977.
32. R.Rivest, "Remarks on a proposed cryptoanalytic attack on the MIT public key cryptosystem", Cryptologia, Jan 1978.
33. H.C.Williams and B.Schmid, "Some remarks concerning the MIT public key cryptosystem", BIT, vol 19, 1979.
34. G.R.Blakley and I.Borosh, "Rivest-Shamir-Adelman public key cryptosystems do not always conceal messages", Comp. and Maths. with Appls., vol 5, 1979.
35. R.G.F. Aitchson, "A cryptographic approach to software", Workshop on electronic privacy and authentication", The Hatfield Polytechnic, July 1982.
36. R.M.F.Goodman and A.J.McAuley, "Broadcast Public Key Cryptosystems", IEEE Int. Symp. on Inf. Th., Les Arcs, France, Jun 1982.
37. R.L.Rivest, "A description of a single-chip implementation of the RSA cipher", Lambda, MIT, 1980.
38. D.Simmons and S.E. Tavares, "An NMOS implementation of a large number multiplier for data encryption systems", IEEE Trans, 1983.
39. A.J.McAuley and N.Parker, "MA743 advanced cipher processor-preliminary data", GEC Research Labs, Hirst Research Centre, 1983.

40. L.N.Lee and S.C.Lu, "A multiple-destination cryptosystem for broadcast networks", COMSAT Tech. Rev., vol 9, no 1, 1979.
41. F.Luccio and S.Mazzone, "A cryptosystem for multiple communication", Inf. Procc. Letts., vol 10, no 4, July 1980.
42. S.T.Kent, "Security requirements and protocols for a broadcast scenario", IEEE Trans. Inf. Th., vol COM 29, no 6, June 1981.

PANEL DISCUSSION

ON

CRYPTOGRAPHY AND SECURITY

Friday 22nd July 1983 at 10.00 hours

Chairman and Organiser:	Professor J.L. Massey	ETH-Zentrum, Zürich, Switzerland.
Panel Members:	Professor I. Ingermarsson	Linköping University, Sweden.
	Dr. S. Harari	Université de Picardie, Amiens, France.
	Dr. R. Johannesson	University of Lund, Sweden.
	Mr. P.G. Wright	Marconi Research Centre, Great Baddow, U.K.
	Dr. R. Blom	Linköping University, Sweden.
	Professor J. Ziv.	Technion - IIT, Haifa, Israel.
	Dr. R.M.F. Goodman	University of Hull, U.K.
Contributors:	Dr. D. Andelman	IAA, Haifa, Israel.
	Dr. B.K. Bhargava	Concordia University, Montreal, Canada.
	Dr. R.E. Blahut	IBM Corporation, Owego, NY, U.S.A.
	Professor K.W. Cattermole	University of Essex, Colchester, U.K.