THE STORY OF NON-SECRET ENCRYPTION by J H ELLIS

1. Public-key cryptography (PKC) has been the subject of much discussion in the open literature since Diffie and Hellman suggested the possibility in their paper of April 1976 (reference 1). It has captured public imagination, and has been analysed and developed for practical use. Over the past decade there has been considerable academic activity in this field with many different schemes being proposed and, sometimes, analysed.

2. Cryptography is a most unusual science. Most professional scientists aim to be the first to publish their work, because it is through dissemination that the work realises its value. In contrast, the fullest value of cryptography is realised by minimising the information available to potential adversaries. Thus professional cryptographers normally work in closed communities to provide sufficient professional interaction to ensure quality while maintaining secrecy from outsiders. Revelation of these secrets is normally only sanctioned in the interests of historical accuracy after it has been demonstrated clearly that no further benefit can be obtained from continued secrecy.

3. In keeping with this tradition it is now appropriate to tell the story of the invention and development within CESG of non-secret encryption (NSE) which was our original name for what is now called PKC. The task of writing this paper has devolved on me because NSE was my idea and I can therefore describe these early developments from personal experience. No techniques not already public knowledge, or specific applications of NSE will be mentioned. Neither shall I venture into evaluation. This is a simple, personal account of the salient features, with only the absolute minimum of mathematics.

4. The story begins in the 60's. The management of vast quantities of key material needed for secure communication was a headache for the armed forces. It was obvious to everyone, including me, that no secure communication was possible without secret key, some other secret knowledge, or at least some way in which the recipient was in a different position from an interceptor. After all, if they were in identical situations how could one possibly be able to receive what the other could not? Thus there was no incentive to look for something so clearly impossible.

5. The event which changed this view was the discovery of a wartime, Bell-Telephone report by an unknown author describing an ingenious idea for secure telephone speech (reference 2). It proposed that the recipient should mask the sender's speech by adding noise to the line. He could subtract the noise afterwards since he had added it and therefore knew what it was. The obvious practical disadvantages of this system prevented it being actually used, but it has some interesting characteristics. One of these, irrelevant to the main theme, is the amusing party trick of using the negative of the speech signal as the added noise. This leaves no signal on the line but the received signal unimpaired. This is easy to do and somewhat startling, but a simple analysis of the feedback shows that it is simply an amplifier with a low input impedance which shorts out the line.

6. The relevant point is that the receiver needs no special position or knowledge to get secure speech. No key is provided; the interceptor can know all about the system; he can even be given the choice of two independent identical terminals. If the interceptor pretends to be the recipient, he does not receive; he only destroys the message for the recipient by his added noise. This is all obvious. The only point is that it provides a counter example to the obvious principle of paragraph 4. The reason was not far to seek. The difference between this and conventional encryption is that in this case the recipient takes part in the encryption process. Without this the original concept is

still true. So the idea was born. Secure communication was, at least, theoretically possible if the recipient took part in the encipherment.

7. The next question was the obvious one, "Can this be done with ordinary encipherment? Can we produce a secure encrypted message, readable by the authorised recipient without any prior secret exchange of the key etc?" This question actually occurred to me in bed one night, and the proof of the theoretical possibility took only a few minutes. We had an existence theorem. The unthinkable was actually possible. The only remaining question was "Can it be made practicable?" This took a while to answer.

8. I published the existence theorem in 1970 (reference 3). Its outline is as follows. We may represent an encipherment process by a look-up table, with the settings, message etc as the variables used for look-up, and the cipher text as the contents of the table. Such a table will normally be impossibly huge but it could, in principle, always be constructed. Conversely, such a table itself can be used as such a process, even if a more conventional embodiment cannot be found. This proof treats the encipherment processes as tables, and demonstrates a form which satisfies the requirements.

9. Suppose the recipient has two tables Ml and M3 while the sender has one, M2. These machine tables are not secret and may be supposed to be possessed by the interceptor. Ml takes an input k and produces an output x. M2 takes inputs x and p giving an output z. M3 takes inputs z and k. All these quantities are large numbers of the same magnitude. We can think of Ml as a linear table or simple list, while M2 and M3 are square tables.

10. In operation p is the message which is to be sent, and k is a random number, the key, chosen by the recipient. He enciphers k by Ml to get x which he sends. The sender uses x to encipher p with M2 to get z, the cipher text, which he sends back. Now the recipient uses k to decipher z by means of M3. It is clearly possible for the entries of M3 to give p under these circumstances, so we have achieved our objective.

11. If the numbers are large enough, and Ml and M2 sufficiently random to avoid working backwards, p cannot be found without knowing k. In public-key-encryption terms, x is the public, encipherment key and k the private, decipherment key.

12. Having thus demonstrated that NSE was possible, the next task was to find a practical implementation. There was no difficulty in getting devices to behave as Ml and M2 in producing output from which the input could not be found, although it was theoretically defined by the output. The problem was to devise a method for which M3 could be produced. Various ideas were found to have flaws, and there was always the possibility that we were trying to break some subtle law of mathematics in looking for practicability. Of course there was no need to use the format of the existence theorem; that was only one option. For instance, suppose we could find two secure encipherment processes which commuted; presumably one process with two different keys but not necessarily. Then the sender (S) could encipher his message, p, and send it to the recipient (R). R could now re-encipher and sent it back to S. Because the two encipherments commute the result would be the same as if R had enciphered first, and S second; so S could remove his encipherment. This text, which he sends back to R is the same as p enciphered by R, so R can decipher. At no time is unenciphered text or key transmitted. This has the disadvantage of an extra pass, but this does not debarr it.

13. Because of the weakness of my number theory, practical implementations were left to others.

The first workable idea was put forward in reference 4 by Clifford Cocks. This is essentially the RSA Algorithm. Briefly the method is this. R chooses two large primes P and Q also prime to P-l and Q-l, and sends N=PQ to S. S has a message which he enciphers as $C=M^N$ (mod N).

14. To decipher, R finds P´ and Q´

$$PP' = 1 \quad (\text{mod } Q\text{-}1)$$

and

$$QQ' = 1 \quad (\text{mod } P\text{-}l)$$

Then

$$M = C^{P'} \quad (\text{mod } Q)$$

$$M = C^{Q'} \quad (\text{mod } P)$$

so M can be found. The security lies in the difficulty of factorising N.

15. It is interesting to note that this method follows exactly the existence theorem; k is P and Q, x is N, p is M and z is C. Ml is the startlingly simple process of multiplying the two parts of k, P and Q, together. M2 consists of raising M to the power N (mod N) and M3 is the process of paragraph 14.

16. The RSA algorithm (reference 5) differs in that R forms a pair of integers, d and e, such that

$$de = 1 \ (\text{mod } (P\text{-}l)(Q\text{-}1))$$

He sends e and N to S, and S enciphers M as

$$C = M^e \ (\text{mod } N)$$

R deciphers as

$$M = C^d \ (\text{mod } N)$$

17. The differences between the two algorithms are superficial. Cocks is a special case of RSA in that it puts e = N. Also it suggests a more computationally efficient method of decryption by retaining the factorisation of N and carrying out the exponentiation twice, but to the smaller moduli P and Q. Of course the decipherment could still be done by forming d such the dN = 1 (mod(P-1)(Q-1)). This technique for reducing the computation is now well-known in the context of RSA.

18. The RSA algorithm has the merit that it is symmetrical; the same process is used both for encipherment and decipherment, which simplifies the equipment needed. Also e can be chosen arbitrarily so that a particularly simple version can be had for encipherment. In this way the complex process would be needed only for the recipient. Indeed, the same simple e could be used for all encipherments, with the recipients just supplying different N. These differences, however are small. The two algorithms are variants of the same method.

19. A couple of months later Malcolm Williamson came to me with a different scheme. This also involved the raising of the message to a power which was the product of large numbers, but

relied on the fact that exponentiation was commutative. He published this in reference 6.

20. The paper was couched in the rather more general form of finite rings; but in terms of modular arithmetic the scheme was for S and R each to choose a large number, k and l respectively. Taking the message again as M, to give some consistency in our notation, S sends M^k and R returns $(M^k)^l = M^{kl}$; all calculations being mod some large prime, say P. S forms K so that Kk = 1 (mod P-1) and therefore $M^{Kk} = M$ (mod P). So that by raising M^{kl} to the power K, S can remove his original encipherment leaving M^l which he sends to R. R removes l in the same way and so recovers M. The fact that k and l need not be prime enormously helps their choice. They must of course, in this version be prime to P-1.

21. Later he produced a simple, elegant scheme which was much easier to use. It was the sort of idea which is obvious once some-one else has thought of it. In this a base x and a modulus q are known. x will be small and q large. S and R each choose a large random number; say a and b. They then form x^a and x^b respectively and send the results to each other. They both now form x^{ab} by raising the number they have received to the power of their own chosen number. Thus after two passes and no decipherment process they both have a large number known only to them which can be used as key in the normal way.

22. I described this method internally on a number of occasions, and, for this reason I suppose, I find I have often been given credit for it although I did, of course, acknowledge the source. I should like to take this opportunity of reaffirming that the credit belongs to Malcolm Williamson. He did publish it, much later than he thought of it, in reference 7. The method was published in reference 8 by Diffie and Hellman. This was identical to Williamson's version, except that they restricted q to be prime.

23. One major class of openly published schemes is based on the Knapsack Problem. The first such scheme to be published was due to Merkle and Hellman (reference 9). This was not anticipated within the closed community, but, although the Knapsack Problem itself is NP complete, the security of schemes based on it is weak.

24. This is the history of invention and early development of NSE by CESG. Some time after the basic work had been done reference 1 was published by Diffie and Hellman. This was the start of public awareness of this type of cryptography and subsequent rediscovery of the NSE techniques I have described.

REFERENCES

1. W Diffie and M E Hellman, Multiuser Cryptographic Techniques, 1976 National Computer Conference, New York City, 7-10 June 1976.

2. "Final Report on Project C43", Bell Telephone Laboratory, October 1944, p. 23.

3. J H Ellis, The Possibility of Secure Non-Secret Digital Encryption, CESG Report, January 1970.

4. C C Cocks, A Note on Non-Secret Encryption, CESG Report, 20 November 1973.

5. R L Rivest, A Shamir, L Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", MIT Laboratory for Computer Science, Technical Memo LCS!TM82, Cambridge, Massachusetts, 4/4/77. Also Comm ACM Vol 21, Feb 1978.

6. M J Williamson, Non-Secret Encryption Using a Finite Field, CESG Report, 21 January 1974.

7. M J Williamson, Thoughts on Cheaper Non-Secret Encryption, CESG Report, 10 August 1976.

8. W Diffie and M E Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, Vol IT-22, No 6 November 1976.

9. R C Merkle and M E Hellman, "Hiding Information and Receipts in Trap Door Knapsacks", presented at the Internal Symposium on Information Theory, Cornell University, Ithaca, New York, October 1977.
Also: IEEE Transactions on Information Theory, Vol IT-24 September 1978, p. 525