AN ALTERNATIVE TO PUBLIC KEY ENCRYPTION

W. M. Hawthorne

United Kingdom

KEY DISTRIBUTION IN AN UNMANAGED NETWORK

Distribution of session keys is the single most important issue in encryption. Since electronic communication is world-wide, the distribution problem is assumed to be complicated and there is already a plethora of elaborate mechanisms for maintaining security, all based on interventionist strategies such as Key Distribution Centres (KDC) and Certification Authorities. Chokhani (1) reports that "Internet, CCITT and ANSI X9.30 have chosen certification", and have even reached the stage of defining mechanisms for a Certificate Issuing Authority to revoke some of the certificates it has already issued! A four layer national Certification Authority is proposed. Neuman and Ts'O (2) advocate the Kerberos Authentication Service which, amongst other things, maintains a database of passwords, issues tickets from a ticket granting exchange, requires a separate authentication exchange and a time-stamping procedure based on a five minute "window".

This paper calls into question whether such elaboration is either necessary or desirable. For at least two thousand years it has been fundamental to the passage of secret information that both the originator's and recipient's prime concern is to exclude third parties. Any form of intervention is therefore a method of last resort. This paper proposes an alternative in which no form of third party management whatsoever is necessary.

Zero Knowledge

The alternative solution is to devise means whereby the originator and recipient, without reference to a third party, exchange open information prior to transmission of the main message from which they and they alone can deduce the session key which will be used to encrypt the message. Systems which provide this facility can be broadly classified as "Zero Knowledge" systems. Within this broad category are Public Key systems which guard the session key by an assymetric encryption method so that the sender uses an encryption key, but, in common with an eavesdropper, has no knowledge of the decryption key The two best known proponents are Diffie and Hellman (3) and Rivest, Shamir and Adleman (4). Diffie-Hellman relies on the inherent difficulty of solving a discrete logarithmic equation, and RSA relies on the difficulty of factoring the product of two primes. Crucial to each method for achieving the required level of cryptographic strength which, in turn, achieves Zero-Knowledge conditions, is the use of very large prime numbers. Neither method inherently authenticates the originator of the message.

THREE REQUIREMENTS OF AN ALTERNATIVE SYSTEM

This paper proposes an alternative system which eliminates the need for large primes. It can be implemented by a compact algorithm and uses small amounts of memory. The system draws on a method of generating a sequence in which the problem of tracing the sequence back to its primitives is immensely difficult. The three essential requirements of the proposed system are

- a "UNIQUE IDENTITY STRING"
- a "UNIQUE CRYPT STRING"
- a KEY MANAGEMENT ALGORITHM

The UNIQUE IDENTITY STRING is randomly generated and buried in a tamper-proof manner within a stand-alone unit or within a transmission device. The present writer has experience of providing such a feature in an authentication device for use by a British bank. At the manufacturing stage the string is burnt into a protected chip. There is no administrative necessity to cross-reference the value of the string with any external number on the cover of the unit, nor is there a requirement for the identity of the string ever to be known for overall management purposes. The UNIQUE CRYPT STRING is installed in a similar way. The complexity of each string is not bound by any mathematical constraint such as is the case in a prime number. There is therefore no upper limit to the strength of the system as a whole. Although both strings require physical protection, all other requirements of the proposed system can be provided in software if appropriate.

'European Convention on Security and Detection', 16–18 May 1995, Conference publication No. 408, © IEE, 1995

The third essential requirement, the KEY MANAGEMENT ALGORITHM (KMA), is in essence a sequence generator of special design capable of accommodating a wide variety of primitives. A preliminary algorithm converts an inputted string to a set of numbers which serve as the primitives for generating a sequence of very large cyclic length. The sequence, known as the MULTI-MODULATED SEQUENCE (MM) is formed by summing separate sub-sequences, each originating from a "PHASE" number which is successively multiplied by a BASE number and modulated by a prime (M). Thus the value of the Phase number changes as the sequence progresses. A typical stream is shown in TABLE 1 in which ten sub-streams (shown in columns) combine to create the first thirteen terms of an MM of cyclic length 7.892 E 41. Extra string variables can add further complexity to the MM. This can be done by concatenation, in which case the extra strings form additional Phase and Base numbers which add to the number of sub-streams and greatly increase the cyclic length. Alternatively, as is shown in the table, complexity can be increased by "fusion" in which existing Phase and Base values are modified without increasing the number of sub-streams. The terms of the MM are normally distributed, but if a stream is formed by modulating each term with a number which is small in comparison to the range of the MM, then such a stream is rectangularly distributed. It also retains the cyclic length of the MM, and is, in effect, pseudo-random. In the table, the MM is modulated by 10 and appears as ES (the encryption stream) and is used to encrypt a "plain" message. It would seem to be a fair assumption that calculating the concatenated string from knowledge of the ES and the fused string is as difficult as the RSA task of factoring the product of two 200-digit primes.

REGISTRATION BETWEEN UNITS

The ultimate use of these three essential requirements is to provide means so that two apparently disparate units with hidden and unknown variables in each can send and receive messages. To achieve this Zero-Knowledge condition, the units must develop some mutual secret so that their activity cannot be known by a third unit. This mutual state of secrecy is achieved in a registration process. It is emphasised, however, that registration between originator and recipient is a brief exchange and is a once-for-all procedure. When completed, all future transmissions require no additional handshaking nor any intervention on the part of the users. In routine use it is therefore, to all intents and purposes, automatic.

The flow diagram for Registration is shown in Figure 1. To underline the practicality of the proposed system, the flow diagram is specific to a fax unit with a UNIQUE IDENTITY STRING and a UNIQUE CRYPT STRING built in at manufacture, communicating with another unit similarly equipped. Since fax machines normally accept dialled numbers rather than strings, the string variables in the flow diagram are numeric and the accompanying numbers represent the number of decimal digits in the variable. Thus, for example, the concatenation of the UNIQUE IDENTITY STRING and the UNIQUE CRYPT STRING forms a 64-digit number.

Registration step by step

The steps in the registration procedure are: The originator at STATION A and the recipient at STATION B agree a ONE-TIME KEY. This agreement is reached in a context off line so that an eavesdropper already on the line cannot benefit. In normal business practice there is ample opportunity for establishing a one-time key in an informal way. In any case, it is inconceivable that two parties who had never established some common ground would ever want to communicate secrets to one another. The UNIQUE IDENTITY STRING is concatenated with the UNIQUE CRYPT STRING and fused with the addresses of A and B to form the primitive Phase and Base values of the KEY MANAGEMENT ALGORITHM (KMA) which is set to +1 (encryption mode), as opposed to -1 (decryption mode). The UNIQUE CRYPT STRING also forms the message, so the structure is irreversible. The output is the MUTUAL PRIMITIVE. As the name implies it is the variable which, when established in the recipient's unit, becomes the common element which allows encryption and decryption to take place. The MUTUAL PRIMITIVE is encrypted by KMA+1, using the ONE-TIME Key as primitive. Since the ONE-TIME KEY is 16-digit it is built up by duplication to 64 digits. This is done at all other points in the algorithm where the primitive falls short of 64 digits. The encrypted mutual primitive is then transmitted to the recipient as the TRANSFER KEY.

The recipient uses the ONE-TIME KEY as primitive and KMA-1 to decrypt the TRANSFER KEY and re-create the MUTUAL PRIMITIVE. The recipient then encrypts the MUTUAL PRIMITIVE using KMA+1 and also using as primitives, his own UNIQUE IDENTITY STRING concatenated to his own UNIQUE ENCRYPT STRING and fused with the addresses of originator and recipient. The output of this process is the REGISTERED CRYPT

STRING, which will be used by the originator in future automatic communication. The REGISTERED CRYPT STRING is sent back in plain to the originator, where it is stored in memory along with the recipient's address. The recipient unit does not store the REGISTERED CRYPT STRING. The entire process is carried out by the machine. The only user task is entering the normal fax numbers and the ONE-TIME KEY.

AUTOMATIC KEY EXCHANGE AND MESSAGE ENCRYPTION

Once registration has been established between two units, the KEY MANAGEMENT ALGORITHM (KMA) provides means for automatic communication. The flow diagram for registration is shown in Figure 2. An important element in this process is the generation of a RANDOM SESSION KEY. This is the secret key which is used in a separate encryption algorithm to encrypt the main message, and which is sent in encrypted form as the second of two headers. It is of course necessary that the recipient can decrypt the header in order to use it to decrypt the main message. This paper is concerned only with key management and not the cryptographic strength of the particular algorithm that is chosen as the means of encrypting the main message. Any symmetrically keyed cipher is suitable. In the interests of generality it is called the MESSAGE ENCRYPTION ALGORITHM (MEA). In a feasibility study for fax carried out by the present writer, it has, however, been found useful to design a carrier cipher which uses the system primes (M) and some of the sub-routines already provided for KMA.

Automatic Key Exchange and Encryption step by step

The originator creates the MAIN MESSAGE. It is the fundamental purpose of the process as a whole that this message arrives securely at the recipient's address. The originator's KMA+1 re-creates the MUTUAL PRIMITIVE. The originator's KMA+1 also creates a RANDOM SESSION KEY appropriate to the design qualities of the MEA. The RANDOM SESSION KEY becomes the primitive to encrypt the MAIN MESSAGE using the MEA+1. The RANDOM SESSION KEY is also encrypted by the KMA+1, using the MUTUAL PRIMITIVE as primitive, to create the ENCRYPTED SESSION KEY. In this latter encryption process a 4-figure randomly generated "open key" is fused with the existing primitives to ensure that the key stream produced is not unvaried. The REGISTERED CRYPT STRING is the first of the two headers sent with the ENCRYPTED MAIN MESSAGE. The recipient's unit decrypts the REGISTERED CRYPT STRING with KMA-1, using as primitives the UNIQUE IDENTITY STRING, the UNIQUE CRYPT STRING and the addresses of recipient and originator, to re-create the MUTUAL PRIMITIVE. The MUTUAL PRIMITIVE then becomes the

TABLE 1 - FORMATION OF A MULTI-MODULATED SEQUENCE

Concatenated string.....
7024556712098772567678903323098776123409096567678332343454 56

Fused string.....	892										
Phase primitives...	704	457	673	211	879	258	769	892	334	311	
Base primitives....	879	614	342	911	658	769	835	236	347	458	
Phase primitives modified by addition of fused string to P(1)...											
	1596	457	673	211	879	258	769	892	334	311	
Modulators...											

31607	31583	31547	31259	31139	30803	30539	30467	30347	30323	MM	ES
12176	27934	9337	4667	17880	13584	796	27710	24857	21146	160087	7
19538	1907	7007	413	25637	3879	23341	19622	6831	11831	120006	6
11301	2327	30369	1135	22947	25863	5853	30275	3291	21104	154465	5
8981	7543	7235	2438	27850	20712	1015	15622	19138	22918	133452	2
24156	20284	13704	1629	15568	2377	22972	285	25240	4686	130901	1
24827	10674	17812	14846	30152	10536	3128	6326	18344	23578	160223	3
14103	16155	3133	20818	4473	995	16065	53	22845	3736	102376	6
6593	2108	30435	22244	16168	25883	7654	12508	6648	13000	143241	1
11166	30992	29807	8452	20145	5289	8439	27056	484	10692	152522	2
16744	16122	4313	10058	21335	1245	22595	17613	16213	14933	141171	1
20721	13429	23884	3951	25880	2512	24262	13156	11716	16639	156150	0
8127	2243	29202	4576	27146	21942	11413	27649	29301	9589	171188	8
451	19133	18232	11289	19421	24157	1687	5226	1202	25250	126048	8

Range of MM = 10 - 309604 **Cyclic length of MM** = 7.892 E 41

primitive to decrypt the ENCRYPTED SESSION KEY using KMA-1 in order to re-create the RANDOM SESSION KEY. The recipient's unit now has the essential information required to decrypt the MAIN MESSAGE, namely, the RANDOM SESSION KEY.

The procedures illustrated by the example of fax are applicable to all branches of open electronic communication where an external system of key management is impractical. By virtue of the registration process, subsequent automatic transmission between any pair of originators and recipients is a unique cryptological path which authenticates both. So even if the main message is sent in plain, a short preliminary encrypted message serves as an effective signature.

1. Chokani,S., 1994, IEEE Communications Magazine, September, 71.

2. Neuman,B.C. and Ts'O,T., 1994, IEEE Comrnunications Magazine, September, 33-38

3. Diffie,W. and Hellman,M.E.,1976,Trans.IEEE on Information Theory, IT-22, No.6,644-654

4.Rivest,R.L.,Shamir,A.and Adleman,L.,1978, Comm.ACM,21,No.2,120-126

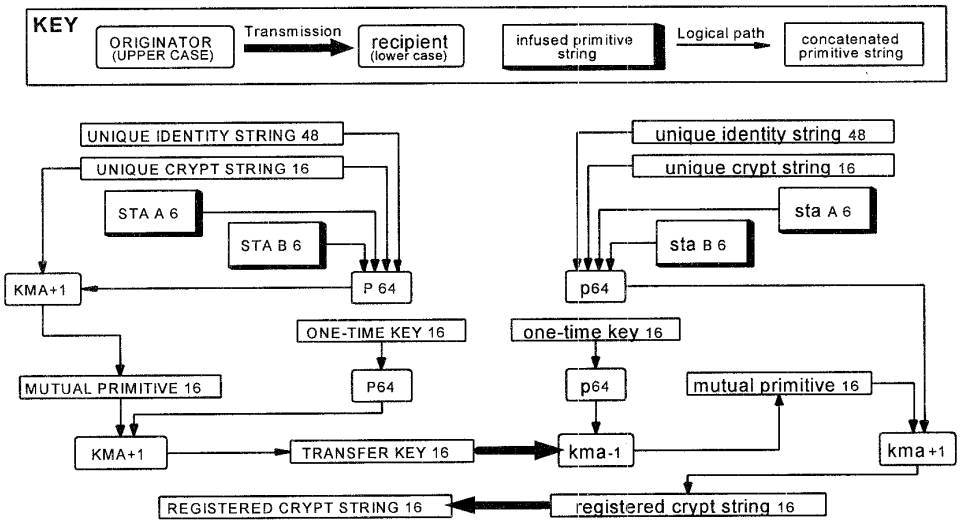

Figure 1 REGISTRATION

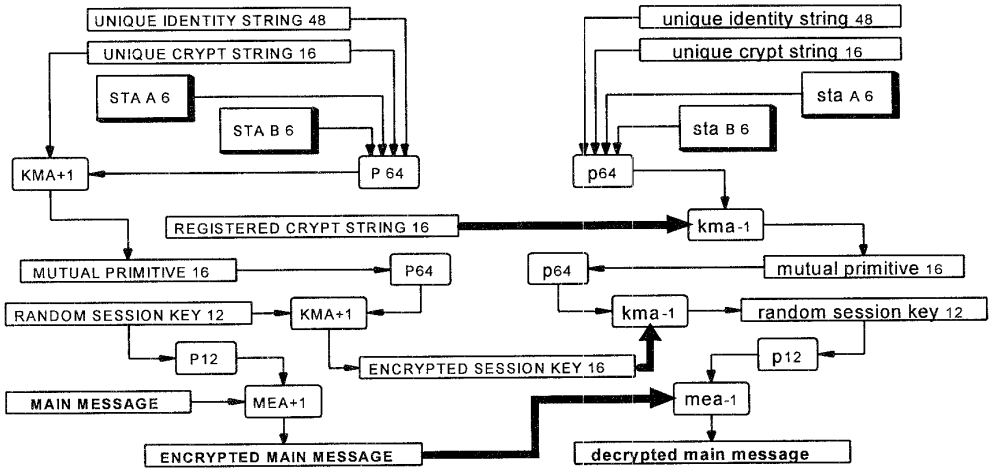

Figure 2 AUTOMATIC KEY EXCHANGE AND MESSAGE ENCRYPTION