[3] J. Cossey, *On metanilpotent Fitting Classes*, Research Report No. 18, 1986, Australian National University, Canberra.

[4] R.S. Dark, *Some examples in the theory of injectors of finite soluble groups*, Math. Zeitschrift 127, 1972, 145-156.

[5] B. Fischer, W. Gaschütz and B. Hartley, *Injektoren endlicher auflösbare Gruppen*, Math. Zeitschrift 102, 1963, 300-305.

[6] T.O. Hawkes, *On metanilpotent Fitting Classes*, J. Algebra 63, 1980, 495-483.

[7] B. McCann, *On Fitting Classes of Groups of Nilpotent Length Three*, Dissertation, Universität Würzburg, 1985.

[8] B.McCann, *Examples of minimal Fitting Classes of Finite Groups*, Archiv der Mathematik,49(3), 1987, 179–186

Department of Mathematics
University College Galway

Asymmetric Cryptography

Patrick Fitzpatrick

1 Introduction

In recent years a great deal of attention has been focussed internationally on the twin problems of *security* and *authentication* in the use of electronic communication systems for a wide variety of transactions including information storage and retrieval, banking and financial transactions and the transfer of legal documents (contracts, invoices etc.). These problems may be summarised as follows:
(a) security – the message must not be capable of interpretation or alteration in any way by an unauthorised person;
(b) authentication — the identities of the parties involved in the communication must be reliably established in such a way that neither can later repudiate any part of the transaction.

The need for cryptographic systems is thus placed firmly in the public domain and is no longer the sole preserve of government, diplomatic and military establishments.

The classical solution to problem (a) is the encryption of messages using a secret key known only to the transmitter and receiver. The key itself must be exchanged by some reliable method — a trusted courier, for instance. However, as the number of participants grows (consider, for example, the national and international branch network of a large banking corporation) the problem of *distribution and secure storage of keys* becomes exceedingly difficult. Moreover, the classical method provides no solution whatever to problem (b).

Since the publication of Diffie and Hellman's fundamental paper [8], it has widely been recognised that *asymmetric* (or *two-key* or *public-key)* cryptosymtems represent *in theory* the best approach towards a solution of these problems. In practice there are few realistic working models — proposed implementatins have either been shown to be insecure or too costly for application in general. As a consequence, a good deal of research has also been devoted to other methods (such as Siegenthaler's work on *stream ciphers* [40, 41]) and, in addition, attempts have been made to apply asymmetric tech-

niques to restricted types of transaction. This latter area is the subject of a major research effort by the EC [31].

The purpose of this article is to review progress in asymmetric cryptography, concentrating on the two principal proposed schemes — *knapsack methods* and *RSA methods*. These topics were the subject of M.Sc. project work carried out at University College, Cork in 1987 and I am grateful to my students Ian Holland, Harry Lande and Michelle Sliney for their endeavours, the results of which can be found in [14], [18] and [44] respectively. This survey owes much to their diligence.

Many readers will also be familiar with the central ideas of asymmetric cryptography. Simmons' Intelligencer article [43] is an excellent introduction (see also Gardner [9]), while Denning [7], DeMillo [6] and Simmons [42] all provide more comprehensive treatments of cryptography and data security. We conclude this introduction with a brief summary of the essence of these ideas.

Both parties to a communication have an *encryption function* E and a *decryption function* D with the following properties:

(i) $D(E(M)) = M$ for every valid message M;

(ii) E and D are easy to compute;

(iii) it is computationally infeasible to determine D from a knowledge of E.

A fourth property which may or may not be present is

(iv) $E(D(M)) = M$ for every valid message M.

The *key distribution problem* is solved when each user places his encryption function in a public file. When user A wishes to communicate "plaintext" M to user B he transmits the "ciphertext" $E_B(M)$ using $B's$ public encryption function E_B. On receipt, B calculates $D_B(E_B(M)) = M$ using his (secret) decryption function D_B. (In practice the actual algorithms used will be known to all parties — including potential intruders. The unknown part is the encryption/decryption key. Here, and throughout the paper we are referring to "algorithm plus key" as the "encryption/decryption function".) The *security problem* is solved provided an intruder can neither interpret $E_B(M)$ — which is the classical requirement — nor tamper with it. Property (iii) is crucial in this regard. Finally, the *authentication problem* is solved in the presence of property (iv), by the following protocol: A sends both $E_B(M)$ and $E_B(S)$

where $S = D_A(M)$; on receipt B computes $D_B(E_B(M)) = M$ and compares it with $E_A(D_B(E_B(S))) = E_A(S) = E_A(D_A(M)) = M$ using A's public encryption function E_A. If these are identical B is assured not only that A is the transmitter (since only A knows D_A) but also that the message sent was in fact M. Thus S is A's (*message dependent*) *signature* appended to the particular plaintext M, so it is ensured that A cannot later deny having sent the message or repudiate any of its content. Acknowledgement by B and message confirmation is required also and it is clear how an independent third party (such as a court of law) can establish the facts of transmission and reception together with the content of the message, in much the same way as is currently the case with paper transactions.

The existence or otherwise of functions satisfying (i) — (iii) or (iv) has not yet been established. Attempts so far have concentrated on the idea of putting some well-known hard problem between knowledge of E and D in such a way that some additional information will allow (computationally) easy access from one to the other. Thus the encryption function is regarded as a *"one-way" function,* that is, a function F for which $f(x)$ is easy to compute for every x, but for which, given y, it is computationally infeasible to determine x such that $f(x) = y$ without some additional *"trapdoor" information*. The two best-known attempts have tried respectively to put the knapsack problem and the integer factorisation problem between E and D. We take these up in turn.

2 Knapsack Methods

The *general knapsack problem* is as follows. Given a set of n postive integer weights $\mathbf{a} = (a_1, \ldots, a_n)$ determine whether a weight N can be obtained by adding together a subset of the given weights, that is, whether there exists a binary vector with n components $\mathbf{m} = (m_1, \ldots, m_n)$ such that $N = \mathbf{m}.\mathbf{a}$. It is well-known (see [10], for example) such that in this generality KNAPSACK is in the *class NP* — a proposed solution $\mathbf{m}$ can be checked in polynomial time, but no polynomial time algorithm is known for determining a solution $\mathbf{m}$ from $\mathbf{a}$ and N. Moreover, KNAPSACK is *NP-complete* so in a sense it is among the most difficult of *NP* problems. However, some instances of KNAPSACK are easy to solve. In particular, if the a_j form a *superincreasing sequence*: $a_{k+1} > \sum_{j=k+1}^{n} a_j$ for $k = 1, \ldots, n-1$ this is clearly the case, since then $m_n = 1$ if and only if $N \geq a_n$ and, for $1 \leq k \leq n-1$, $m_k = 1$ if and only if

$N - \sum_{j=k+1}^{n} m_j a_j \geq a_k$.

The original knapsack cryptosystem proposed by Merkle and Hellman [28] uses the following scheme. Select a superincreasing sequence $\mathbf{a}'$ and two positive integers P, Q such that P is invertible modulo Q and $\sum_{j-1}^{n} a_j' < Q$. Define $a_j \equiv a_j' P \pmod{Q}$ for all j. Now given the (binary encoded) message block $\mathbf{m}$ of length n, transmit $N = \mathbf{m}.\mathbf{a}$. Here $\mathbf{a}$ is the public-key part of the system. Only the receiver, who knows the secret trapdoor information (P, Q), can compute

$$\begin{aligned} N' &\equiv P^{-1}N \equiv P^{-1}(\sum m_j a_j) \\ &\equiv P^{-1}(\sum m_j a'_j)P \equiv \sum m_j a'_j \pmod{Q} \end{aligned}$$

and since $\sum a'_j < Q$, $N' = \sum m_j a'_j$. This is easily solved since $\mathbf{a}'$ is superincreasing. Note that this algorithm can be used for either security or authentication but not both because property (iv) does not hold — many plaintexts M are not valid ciphertexts so $E(M)$ cannot be calculated.

In 1980, Shamir and Zippel [39] showed that the basic Merkle-Hellman scheme — henceforth referred to as MH — could be broken "almost certainly" if the modulus were known to the cryptanalyst. Later Shamir [37] described a method by which MH could be broken "with high probability" in polynomial time. The essential point in his argument is that there are usually many so-called *trapdoor pairs* (P_0, Q_0) any one of which has the property that $\mathbf{a}'P_0 \pmod{Q_0}$ is superincreasing and gives the correct decryption of the ciphertext. He reduces the search for one of these pairs to a system of linear inequalities in several variables — arguing that four suffice in almost all cases — which he then solves using Lenstra's integer programming algorithm [21].

Merkle and Hellman also suggest in [28] that iterating their basic scheme could lead to improved security. However, building on Shamir's work, Adleman [1] (see also [3]) was able to demonstrate how to break the iterated system. He uses the *"lattice reduction"* algorithm of Lenstra, Lenstra and Lovasz [22] to convert a system of nonlinear equations — under some plausible hypotheses — to a system of linear inequalities and then uses Shamir's approach. He does not prove rigorously that his method works and extensive computer calculations were required to verify that it does so "with high probability in almost all cases" (see [3] for references and further details).

Several other variants of MH are known. Merkle and Hellman [28] themselves suggest a multiplicative version as follows:

Choose n relatively prime numbers $b_1, \ldots, b_n$, a prime p such that $p > b_1 b_2 \cdots b_n$ and a primitive root c modulo p. Determine a_j such that $b \equiv c^{a_j} \pmod{p}$ and use $\mathbf{a} = (a_1, \ldots, a_n)$ as the public key, keeping c and p secret. To transmit the (binary encoded) message $\mathbf{m} = (m_1, \ldots, m_n)$ calculate $k = \sum_{j=1}^{n} m_j a_j$ and send k. The receiver, knowing c and p, can find $m \equiv c^k \pmod{p}$ and since $m \equiv c^k \equiv \prod c^{m_j a_j} \equiv \prod b_j^{m_j} \pmod{p}$ and $p > \prod b_j$ then $m = \prod b_j^{m_j}$ so $m_j = 1$ if and only if $b_j \mid m$. The intruder must either find the m_j knowing only k and $\mathbf{a}$ or else find c and p. The latter brings in the well-known hard problem of computing logarithms in a finite field $(\mathbf{Z}_p)$. This scheme was successfully attacked by Odlysko [30] under the assumption that some of the b_j were known (indeed practical constraints would probably require them to be small — the first few primes, for example), and later by Adleman in more generality (see [3]).

It was clear from the start that a possible source of crytographic weakness in MH lay in the fact that the early knapsack weights in the superincreasing sequence would be significantly shorter (in binary length) then the later ones. Graham (see Lempel [20]) and Shamir [39] independently described another variant of MH in which they disguised the superincreasing structure by "padding" the weights before the modular multiplication so that they all had approximately the same length. This Graham-Shamir scheme has been attacked in certain cases — Brickell and Simmons [3] give the details — using methods similar to those of Adleman. incidentally, at the same time Odlysko [30] successfully attacked the method outlined by Shamir [36] for using the knapsack scheme for signatures instead of security.

In more recent developments Shamir [38] (see also Willett [46]) has described an iterated knapsack cryptosystem which starts from an arbitrary initial knapsack and thus avoids introducing the superincreasing structure. Also, Goodman and McAuley [13] have developed a knapsack based method which brings in the integer factorisation problem. To our knowledge neither of these methods has been cryptanalysed.

3 RSA Methods

The original RSA system (Rivest, Shamir and Adleman [34]) is probably the best known feature of the asymmetric cryptography literature. Given $n = pq$ where p and q are prime the Euler phi function of n is $\varphi(n) = (p-1)(q-1)$. If $(e, \varphi(n)) = 1$ and d is chosen so that $ed \equiv 1 \pmod{\varphi(n)}$ then a message

m (an integer between 0 and $n-1$) can be encrypted as $c \equiv m^e \pmod n$ and decryption is described by the congruence $c^d \equiv m^{ed} \equiv m^{1+k\varphi(n)} \equiv m \pmod n$ since $m^{k\varphi(n)} \equiv 1 \pmod n$ for any k. The public key is (e, n) and the private key is d. (It is clear that the selection of e, d works when $(m, n) = 1$ and easy to see that the equations hold also when p or q — but not both, of course — divides m.)

The essential feature of the method is that there is no known way using present day technology of factoring integers with about 200 decimal digits in any reasonable time. Recent work at the Sandia Laboratories [5] using a CRAY I computer is based on the "quadratic sieve" algorithm of Pomerance [32] and focuses on numbers with between 65 and 100 digits. Other fast factoring algorithms are due to Morrison-Brillhart [29] and Schroeppel (see [35]) with running times for factoring a 150-digit number of about 9×10^8 years and 2×10^{13} years respectively. Since no-one has yet been able to find a way of breaking the RSA scheme which does not involve factoring the modulus (or determining $\varphi(n)$ — it is easy to show this amounts to the same thing), the security is, at present therefore, very high and could be increased if necessary simply by increasing the lengths of p and q (but see the conclusion). Also since the RSA is commutative in the sense of property (iv), it can be used for authentication as well as security. On the other hand, the operation of modular exponentiation is very slow and leads to a throughput rate for the data which compares unfavourably with competing methods (such as stream ciphers or conventional ciphers like the Data Encryption Standard [4]).

As a consequence, the greatest efforts — apart from trying to break the RSA without factoring the modulus — have gone into trying to improve the speed of the algorithms used in its initialisation and implemention: random number generation, primality testing, determination of greatest common divisor and modular multiplication and exponentiation. The primality test suggested in [34] is the probabilistic one of Solovay and Strassen [45] although the OSIS report [31] claims that the test given by Knuth [15 p.379] is provably better. Of course the primes used should in some sense be randomly chosen — several good pseudo-random number generators are known (see, for example Golomb [12]), but the latest work [31] suggests using some physical process (such as heat, white noise or radioactive decay) as a source of truly random numbers. Finally, algorithms for calculations like the GCD and modular arithmetic are constantly being refined and improved (Blakley [2], for instance). Recent work by Kung and his associates on *systolic algorithms* and the corresponding computer architecture in providing a new and exciting

stimulus in this field [16, 17].

In a more general context the RSA may be regarded as using the polynomial function $g_e(x) = x^e$ to permute the elements of the ring $\mathbf{Z}_n$ (where $(e, \varphi(n)) = 1$). Lidl [23] and Lidl and Müller [24] consider other possible *"permutation polynomials"* for use in RSA-type cryptosystems. One such class of functions is the set of *Dickson polynomials* (or *Chebyshev polynomials of the first kind)* defined by

$$g_e(a, x) = \sum_{j=0}^{e/2} \frac{e}{e-j} \binom{e-j}{j} (-a)^j x^{e-2j} \quad (\text{for } a = \pm 1)$$

(when $a = 0$ we recover the RSA polynomial). In [24], it is shown that $(g_u \circ g_v)(a, x) = g_{uv}(a, x) = (g_v \circ g_u)(a, x)$ and Lausch et al [19] prove that $g_e(a, x)$ induces permutation of $\mathbf{z}_n$ with $n = pq$ and p, q prime if and only if $(e, (p^2-1)(q^2-1)) = 1$. Also, g_v is the inverse of g_u if and only if $uv \equiv= 1 \pmod{(p^2-1)(q^2-1)}$ which means that in using these polynomials for cryptosystems the inherent difficulty of factoring n is again brought into the calculation of the inverse v.

Further generalisations are possible using (Chebyshev) polynomials in several variables (see [24]) or rational functions (Rédei [33]) to induce permutations on $\mathbf{Z}_n$. To our knowledge none of these polynomial generalisations of the RSA has actually been analysed as part of a practical cryptosystem.

4 Conclusion

It is somewhat surprising that only a few proposals have been made for algorithms to implement asymmetric cryptosystems. In fact, apart from those mentioned above (and various short-lived variations — see, for example [26], [11]) only one other has been given, namely, a suggestion by McEliece [27] (see also [25] p.360) that *error-correcting Goppa codes* be used — the data is transmitted with many errors which only the recipient knows how to correct. In practice the Merke-Hellman scheme has never been used and the Graham-Shamir system has only been used briefly (by Western Electric), while the RSA was adopted by several groups and implemented on LSI chips at MIT (by Rivest et al) and at Sandia National Laboratories. As mentioned in the introduction the RSA is now the cryptosystem of choice by the European working group OSIS in the design of a secure *"token"* based payment and

financial transfer protocol. These implementations represent, however, only a small minority of the current applications of cryptography. Conventional cryptosystems such as the Data Encryption Standard or DES [4] are in use to a much greater extent and this reflects both a lack of confidence in asymmetric techniques together with the relative inefficiency of the RSA method.

Until now the approach to the design of asymmetric cryptosystems has been to take some known hard problem and build it into the derivation — without trapdoor knowledge — of the content of the message and the decryption function from knowledge of the ciphertext and the encryption function. Thus solving the hard problem implies breaking the cryptosystem and *it is hoped* that the converse is also the case, that is, that the cryptosystem cannot be broken *without* solving the hard problem. In no case has this been proved and, of course, as Shamir and others have amply demonstrated, breaking the knapsack cryptosystems so far proposed is *not* equivalent to solving KNAPSACK in polynomial time.

Thus there remains the underlying doubt as to whether any proposed scheme is secure and whether it will continue to be so into the future. But, in addition, there is he even more fundamental question: Do there exist genuine asymmetric cryptographic functions? Simmons [43] calls this "one of the most important questions in contemporary applied mathematics".

References

[1] Leonard M. Adleman, *On breaking the iterated Merkle-Hellman public-key cryptosystem*, Advances in Cryptology (ed. D. Chaum), 303-308 (Plenus 1985).

[2] G.R. Blakley, *A computer algorithm for calculating the product AB modulo M*, IEEE Trans. C-32, 497-500 (1983).

[3] Earnest F. Brickell and Gustavus J. Simmons, *A status report on knapsack based public-key cryptosystems*, Congressus Numerantium, 37, 3-72 (1983).

[4] *Data Encryption Standard (DES)*, National Bureau of Standards Publ. FIPS-PUB-46 (1977).

[5] James A. Davis and Diane B. Holdridge, *Factorization using the quadratic sieve algorithm*, Sandia Report SAND 83-1346 (1983).

[6] R. DeMillo et al, *Applied Cryptography, Cryptographic Protocols and Data Security*, Proc. Symp. in Appl. Math. 29, (AMS short course lecture notes) 1983.

[7] D.E. Denning, *Cryptography and Data Security*, Addison-Wesley 1982.

[8] W. Diffie and M.E. Hellman, *New directions in cryptography, IEEE Trans.* IT-22, 644-654 (1976).

[9] M. Gardner, *Mathematical Games (section)*, Sc. Amer. 237, 120-124 (1977).

[10] M. Garey, and D. Johnson, *Computers and Intractibility*, Freeman 1979.

[11] J-M. Goethals and C. Couvreur, *A cryptanalytic attack on the Lu-Lee public-key cryptosystem*, Philips J. Res. 35, 301-306 (1980).

[12] S.E. Golomb, *Shift Register Sequences*, Holden-Day 1967.

[13] R.M.F. Goodman, and A.J. McAuley, *New trapdoor-knapsack public-key cryptosystem*, IEE Proc. 132, 289-292 (1985).

[14] Ian M. Holland, *The RSA cryptosystem: Implementation considerations*, M.Sc. Project in Inf. Th., University College, Cork (June 1987).

[15] Donald E. Knuth, *The Art of Computer Programming Vol. 2 – Seminumerical Algorithms (2nd ed)* Addison-Wesley 1981.

[16] H.T. Kung, *Why systolic architectures?* Comp. Mag. 15, 37-46 (1982).

[17] H.T. Kung and R.P. Brent *Systolic VLSI arrays for polynomial GCD computation*, Technical Report Carnegie-Mellon University Comp. Sc. Dept. (May 1982).

[18] Henry A. Lande, *A survey of the RSA cryptographic scheme and related subjects*, M.Sc. Project in Inf. Th., University College Cork (June 1987).

[19] H. Lausch, W. Müller, and W. Nöbauer, *Uber die Struktur einer durch Dicksonpolynome dargestellten Permutations gruppe des Restklassenringes modulo n*, J. Reine Angew. Math. 261, 88-99 (1973).

[20] Abraham Lempel, *Cryptology in transition,* ACM Computing Surveys 11, 285-303 (1979).

[21] H.W. Lenstra Jr., *Integer programming with a fixed number of variables*, Math. of Op. Res. 8, 538-548 (1983).

[22] A.K. Lenstra, H.W. Lenstra Jr. and L. Lovasz, *Factoring polynomials with rational coefficients*, Math. Ann. 261, 515-534 (1982).

[23] R. Lidl, *On cryptosystems based on polynomials and finite fields*, Proc. EUROCRYPT 84, Lecture Notes In Comp. Sc. 209,10–15 (1985).

[24] R. Lidl and W. Müller, *Permutation polynomials in RSA cryptosystems*, Advances in Cryptology (ed. D. Chaum), 293-301 (Plenum 1984).

[25] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, C.U.P. 1986.

[26] S.C. Lu and L.N. Lee, *A simple and effective public-key cryptosystem*, Comsat Tech. Rev., 15-24 (1979).

[27] R.J. McEliece, *A public-key cryptosystem based on algebraic coding theory*, DSN Progress Rep. 42-44, Jet Propulsion Lab. (1978).

[28] R. Merkle and M. Hellman, *Hiding information and signatures in trapdoor knapsacks*, IEEE Trans IT-24, 525-530 (1978).

[29] M.A. Morrison and J. Brillhart, *A method for factoring and the factorization of F_7*, Math. of Computation 29, 183-205 (1975).

[30] Andrew M. Odlysko, *Cryptanalytic attacks on the multiplicative knapsack cryptosystems and on Shamir's fast signature scheme*, IEEE Trans. IT-30, 594-601 (1984).

[31] *Open Shops for Information Services*, OSIS European Working Group Final Report 1985-05-21 (1985).

[32] C. Pomerance, *Analysis and comparisn of some integer factoring algorithms*, Number Theory and Computers (ed. H.W. Lenstra Jr. and R. Tijdeman) Math. Centrum Tracts 154 (1978).

[33] L. Rédei, *Uber eindeutig umkehrbare Polynome in endlichen Körpern*, Acta Sci. Math. (Szeged) 11, 85-92 (1946).

[34] R. Rivest, A. Shamir and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM 21, 120-126 (1978).

35] J. Sattler and C.P. Schnorr, *Ein Effizienzvergleich der Faktorisierungsterfahren von Morrison-Brillhart und Schroeppel*, Computing 30, 91-110 (1983).

[36] A. Shamir, *A fast signature scheme*, MIT Lab. for Comp. Sc. Rep. TM-107 (July 1978).

[37] A. Shamir, *A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem*, Proc. IEEE Symp. Found. Comp. Sc. 145-152 (1982).

[38] A. Shamir, *Embedding cryptographic trapdoor in arbitrary knapsack systems*, Inform. Process Lett. 17, 77-79 (1983).

[39] A. Shamir and R.E. Zippel, *On the security of the Merkle-Hellman cryptographic scheme*, IEEE Trans. IT-26, 339-340 (1980).

[40] T. Siegenthaler, *Correlation immunity of nonlinear combining functions for cyptographic applications*, IEEE Trans. IT-30, 776-780 (1984).

[41] T. Siegenthaler, *Decrypting a class of stream ciphers using ciphertext only*, IEEE Trans. C-34, 81-85 (1984).

[42] G.J. Simmons (ed.), *Secure Communications and Asymmetric Cryptosystems*, AAAS Selected Symposia Series, Westview Press (1982).

[43] G.J. Simmons, *Cryptology: The Mathematics of secure communication*, Math. Intelligencer 1, 233-246 (1979).

[44] Michelle Sliney, *Trapdoor knapsack public-key cryptosystems*, M.Sc. Project in Inf. Th., University College Cork (June 1987).

[45] R. Solovay and V. Strassen, *A fast Monte-Carlo test for primality*, SIAM J. Comp. 6, 84-85 (1977).

[46] M. Willett, *Trapdoor knapsacks without superincreasing structure*, Inform. Process Lett. 17, 7-11 (1983).

Department of Mathematics
University College Cork