Public key cryptography

NIEUW ARCHIEF VOOR WISKUNDE (4), Vol. 1, (1983), 259-269

PUBLIC KEY CRYPTOGRAPHY

J.H. van LINT

In this talk a new and important area of mathematical research will be described. The challenging problems of this area have attracted many researchers in recent years. Only a few years ago it started to play (a still modest) rôle in our research and educational program. The subject of this talk is cryptography, more specifically the so-called *public key cryptography*.

When one hears the word cryptography one usually thinks of the classical model of communication with secret codes, which is described in Figure 1.

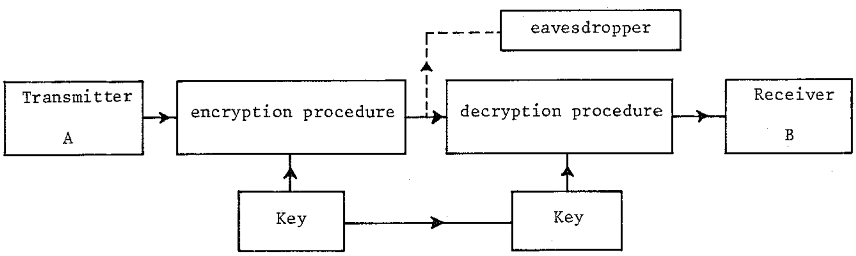

Figure 1.

The conventional example is communication between military units or diplomatic services. Here K_1 is the communication channel which is insecure, i.e., unauthorized parties (eavesdroppers) can extract information from the channel. The information is transmitted over the channel in encrypted form. The procedures used to encrypt resp. decrypt the message depend on a *key* which the transmitter A sends to the receiver B over a secure channel K_2 (e.g., by courier). The problem, which the eavesdropper (usually called a cryptanalyst) must solve, is breaking the code, i.e., he tries to discover the key. Much mathematical work has been done in this area and several impressive results

* This is a slightly modified and translated version of an invited address held for a general audience on the occasion of the 27th 'dies natalis' of the Eindhoven University of Technology.

achieved during world war II are well known (cf. [6]).

A rather simple example of a classical cryptographic system is the *mono-alphabetic substitution* in which a permutation of the alphabet is used. For example

ABCDEFGHIJKLMNOPQRSTUVWXYZ
THEKRZWDLPOBJQVAMIYCSXGUNF

is a key which takes the plaintext word DIES into the ciphertext word KLRY. It is easy to break this system by a statistical analysis of the language which is used. We do not discuss further details of classical cryptography (cf. [5] Chapt. IV, [6]). At the end of the talk the American cryptosystem DES will be mentioned. It is an example of a mono-alphabetic substitution, not on an alphabet of 26 symbols but one with 2^{64} symbols.

New applications

The work which I would like to discuss became necessary for several reasons:

(i) a large number of telephone conservations is sent via satellites nowadays by microwave radio. It has become easy for eavesdroppers to hear these conversations too. It is well known that at least one foreign embassy in Washington listens in on American telephone conversations and that all telephone conversations, telex and telegram messages going to or from the USA are monitored by the National Security Agency (NSA);

(ii) Electronic fund-transfer between banks and business communications by teleprocessing systems are becoming more and more common. In these situations authentication of the source of a message is essential. At present the validity of messages is guaranteed by signatures. What is needed is a digital equivalent of a signature. We shall define such a signature as a message which can be produced by one source only but such that anyone can check the authenticity of this message.

(iii) For many multiuser computer systems it has become necessary to check the identity of a user via a so-called 'login' procedure. This prevents unauthorized use of the computer and information stored in the memory. The problem is to prevent the theft of the passwords used for this login procedure from the memory of the computer.

Since it is becoming increasingly common for information to be transmitted or stored in digital form it is also becoming easier for eavesdroppers to have this information analyzed by (ever faster) computers.

Another problem which arises in modern forms of communication is the introduction by adversaries of false information into the channel, e.g., a repetition of a message which was intercepted earlier.

It is not difficult to understand that cryptography will play an important rôle in the future. We can expect the advent of commercial '*crypto-networks*', which could have thousands of subscribers. It is impossible to let every pair of potential users of the system agree on a separate private key and it will often happen that one wishes to communicate with a subscriber with whom one has had no prior acquaintance. It is unrealistic to assume that such a pair of users could wait until a key is exchanged over some secure channel (such as registered mail).

This introduction should be sufficient for the audience to appreciate the topic of this talk, i.e., the idea of public key cryptosystems, introduced in 1976 by W. DIFFIE and M. HELLMAN [2].

Trap-door one-way functions

The main element in the systems under consideration are the so-called trap-door one-way functions. First, we shall introduce the concept of a one-way function. We consider a function $f: X \to Y$ and for the sake of simplicity we assume that f is one-to-one. We require that f is a 'simple' function. By this we mean that for each $x \in X$ it is easy to compute the value $f(x)$. E.g., a computer program of a few hundred instructions which calculates $f(x)$, given x, is a simple function. Next, we require that, for almost all $y \in Y$, it is computationally unfeasible to solve the equation $y = f(x)$. E.g., a program which calculates $\overleftarrow{f}(y)$ might require 10^{10} instructions, thus taking a computer hundreds of years to execute. We stress that it is not impossible to find the inverse $\overleftarrow{f}$ but that it is computationally unfeasible because too much time (or memory) is necessary. Such a function is called a one-way function.

Let us first look at an application of these functions. We consider a multiuser computer with a login procedure where each user must enter his own secret password x before he gets access to the computer. If the list of users with their passwords is stored in the computer it could come into the wrong hands (e.g., via malevolent system operators). In modern computer systems the computer has the program for a one-way function f and a list of users with the value of f(x) if x is the users password. When the user enters his password x, the computer calculates f(x) and compares it with the stored value. If someone steals the list, then this is of no use to him, even if he also knows the function f, since it would take much too long to first calculate $\overleftarrow{f}$ in order to obtain the passwords.

Now, we come to the idea of the trap-door. If, for a one-way function f, it becomes easy to compute $\overleftarrow{f}$ once certain extra (trap-door) information is known, then we call f a trap-door one-way function.

Below, we shall consider some examples.

Public Key Cryptosystems

A public key cryptosystem works as follows. Let X be the set of possible messages and Y the set of encrypted messages. Each user, say A, determines a pair of one-to-one functions E_A, D_A. Here E_A (encryption) is a trap-door one-way function from X to Y and D_A(decryption) is the inverse function, which A can easily compute since he has the required extra information about E_A. We repeat that knowledge of E_A alone is in principle enough to calculate D_A, but in practice it is not feasible.

The surprising new feature of this kind of system is that each user places his encryption procedure E_A in a *public* directory (with his name and address). Of course each user keeps his decryption procedure D_A secret.

Suppose that user A wishes to send a message x to user B. He then looks up the procedure E_B (the encryption procedure of the receiver!) in the directory and then transmits the message $y = E_B(x)$. The receiver B computes $D_B(y) = D_B(E_B(x)) = x$. An eavesdropper who intercepts the message y can easily look up E_B in the directory but again, this is of no use to him since the calculation of D_B takes him too long.

The fact that the functions E_A, E_B, ... are one-to-one allows us to introduce the digital signature feature. This works as follows. First A sends his name and address to B, without encryption. Now B knows that he can expect a message from A. Then A changes the message x into $D_A(x)$, using the function D_A which is known only to A. He then transmits $y = E_B(D_A(x))$. (Here x must be restricted s.t. $D_A(x)$ belongs to the domain of E_B). As before, the receiver can calculate $D_A(x)$, using his own decryption function D_B. He stores the message $D_A(x)$. Next, B looks up the function E_A in the public directory and then calculates $E_A(D_A(x)) = x$. The receiver B is now sure that the message was sent by A. Furthermore A cannot deny having sent the message because B has saved $D_A(x)$ and anybody can check that $E_A(D_A(x)) = x$ but nobody but A could have produced the message $D_A(x)$. Notice that this satisfies our definition of a signature.

For many cryptosystems which have been suggested it is difficult to derive a procedure for a signature. This is one of the areas in which much research is taking place.

The idea of public key cryptography is obviously very nice but it can only be of practical use if we can construct the necessary one-way functions. In 1978 R. RIVEST, A. SHAMIR and L. ADLEMAN [14] designed a system which is now known as the RSA-system (or MIT-system, referring to the inventors' affiliation).

The RSA-system

We consider the messages represented as positive integers x (with many digits), say $0 \le x < n$. Here n depends on the user A, who proceeds as follows:

(1) Determine two large primes p and q (each with about 100 digits).

(2) Take $n := pq$ and $m := \phi(n) = (p-1)(q-1)$.

(3) Choose $d > \max\{p,q\}$ at random from the integers in $[1,m]$ relatively prime to m.

(4) Use Euclid's algorithm to determine e such that $ed \equiv 1 \pmod m$.

The public key for A is the pair (n,e) with encryption procedure $E_A(x) := x^e \pmod n$. The procedure which A keeps secret is $D_A(y) := y^d \pmod n$. Since $ed \equiv 1 \pmod{\phi(n)}$ it is a direct consequence of Fermat's theorem that $D_A(E_A(x)) = E_A(D_A(x)) = x$.

An eavesdropper is faced with the following problem. He knows n and e and he must calculate d. In order to do this he must first calculate m (by (4) above) and by (1) this involves factoring n into the product pq. The problem of factoring integers has interested many number-theorists for centuries. Several algorithms are known. The fastest of these, due to R. Schroeppel, needs more than 10^{23} operations to factor a 200-digit number n. Some day there may be computers that can do this computation in a hundred years, hardly a consolation for the eavesdropper.

However, we should realize that there may be an organisation, for which secret communication is important, which has found a much faster algorithm for factoring integers. In that case such an organisation will make sure that this fact does not become known. At the moment the results of research activities at universities find their way into the literature but even that may change as we shall see below.

The publication of the RSA-system caused several sensational headlines such as : "The new unbreakable codes - will they put NSA out of business?". Of course, several researchers tried to break the RSA-system without resorting to factorization. Some of these attempts were partially successful (cf. [17]) but with a few extra conditions on the choice of p and q it seems that the RSA-system is still completely safe. In fact, it is being used commercially already.

We have not discussed the amount of work which the user A must do. This should take very little time, certainly in those situations where one changes the keys E_A, D_A regularly (for extra security). The calculations (2), (3), (4) are all easy but how does one find a 100-digit prime number? (Shortly after the RSA-system became publicly known an American company offered such primes for sale for a few hundred dollars. Of course, buying primes from others makes security dubious.) If we use a random generator to make a 100-digit odd number, then we have about 1% chance that it is prime; (the reader can check this using the prime number theorem). If it is possible to check in a short time whether a given integer is prime, then the random generator will not take long to produce the pair {p,q}. The subject of primality testing has made great progress in recent years, stimulated by the applications in cryptography. For interesting surveys we refer to [13], [15]. The fastest general primality test which is known at present is due to H. Cohen and H.W. Lenstra, Jr. Their method is a significant improvement of a method developed in 1982 by Adleman, Pomerance and Rumely, which is one of the few results in mathematics that received attention in the press. The Cohen-Lenstra algorithm takes about 45 seconds on the CDC-computer at the SARA-computing center in Amsterdam to check whether a 100-digit number is prime. [18].

The trap-door knapsack-system

Our second example of a pubic key cryptosystem illustrates several aspects of recent research in this area. The system was introduced by R. MERKLE and M.E. HELLMAN [11]. It is based on a well known combinatorial problem, called the knapsack problem. If A is a set of integers and if we wish to calculate the sum S of the elements in a specified subset of A, then this is a simple addition. However, if S is given and we must find the corresponding subset of A, this is in general quite difficult. How this idea is used is illustrated by the following extremely oversimplified example. We represent A in some order as $(a_1, a_2, \ldots, a_n)$ and consider a binary message of length n as a characteristic function of a subset of A. The encrypted message is S.

(i)	public key	A :	3	5	10	22	43	90	201	
	message		1	0	1	0	0	1	0	
	encrypted message	S = 3 + 10 + 90 = 103								
(ii)	public key	A':	130	49	98	115	19	379	159	
	message		1	0	1	0	0	1	0	
	encrypted message	S' = 607 = ? + ... + ?								

Example (i) is trivial because the sequence A is superincreasing, i.e. each a_i is more than the sum of the previous elements. Therefore, given the sum 103, it is obvious that the element 90 was used, etc..

Although the second example is still fairly easy, it looks considerably more difficult than the first. The same message yields the encrypted version 607 and it takes a little reflection to recover the message. We are looking at the problem from the point of view of the eavesdropper. The user who published his public key A' has trap-door information, namely that he constructed A' by multiplying the elements of A by 211 and reducing mod 503. He also knows that $211.267 \equiv 1 \pmod{503}$. Hence he can transform the second problem back into the trivial first example.

Now, let us consider the same idea as it is used in practice. Let $n = 100$ and for $j = 1, 2, \ldots, n$ choose a_j at random from the integers between $(2^j - 1)\, 2^{100} + 1$ and $2^{100+j-1}$. This gives us the 'easy' set A. Next, choose m at random between $2^{201} + 1$ and $2^{202} - 1$, then $\hat{w}$ at random between 2 and $m - 2$, and finally define $w := \hat{w}/(m, \hat{w})$. Then $(w, m) = 1$ and we can calculate $w^{-1} \pmod m$, which is kept secret. The public key consists of the integers $a_i' := wa_i \pmod m$. A binary message $(b_1, b_2, \ldots, b_n)$ is encrypted as $S = \Sigma_{i=1}^{n} a_i' b_i$. The way in which the public key is constructed has the effect that the sequence a_i' $(1 \le i \le n)$ has the appearance of a randomly selected set of large integers.

The security of this system is based on the fact that the knapsack-problem is a hard problem to solve. What does this mean? In order to define what we mean by a hard problem we would have to discuss another fairly young branch of mathematics, namely the analysis of algorithms and *computational* *complexity theory*. In this theory a number of computational problems including the knapsack problem have been shown to be of comparable difficulty (cf. [3]). This class is known as NP (nondeterministic, polynomial). These problems cannot be solved in a time which is polynomial in the parameters of the problem by presently known algorithms unless the computer has an unlimited degree of parallelism. So, the knapsack system looks safe. However, we should be careful. The sequence $a_i'(1 \le i \le n)$ may look like a randomly chosen sequence but we know that it was obtained from a super-increasing sequence, i.e., we know that trap-door information exists. Maybe this type of knapsack problem does not belong to NP. Indeed, on May 12, 1982 there was an article on page 1 of the Los Angeles Times with the headline "Unbreakable Computer Code proves otherwise". A. Shamir had found a fast way to break the knapsack code (cf. [16]). Instead of tackling the knapsack problem itself he had solved the problem: Find a pair w^{-1},m such that the sequence $w^{-1}a_i'(1 \le i \le n)$ reduced mod m is super-increasing (given that at least one such pair exists). Shortly after that, the Director of NSA, admiral B. Inman, stated that NSA had found the idea of public key cryptography several years earlier than Diffie and Hellman and furthermore also the knapsack system. He also claimed that they had discovered that it was easy to break but that NSA saw no reason to make these facts publicly known.

Cryptography and NSA

The remarks above introduce the last topic of this lecture, the many problems which arose in the past few years in connection with research in cryptography, many of which made headlines. As a first example I mention the official American cryptosystem "Data Encryption Standard" (DES) which was adopted by the National Bureau of Standards in 1977. (cf. [5], Chapt. VIII). The system is a transformation of 64-bit data blocks into others, depending on a 56-bit key. In fact, it is a simple mono-alphabetic substitution on an alphabet of 2^{64} symbols. The whole system fits on one LSI-chip and several manufactures of electronic equipment incorporate it. This sounds very nice but something peculiar is going on. The DES was developed by NBS and IBM and the system was proposed in 1975. It immediately stirred up a heated controversy. There were two important criticisms (cf. [12]). First of these is the fact that the key size (56 bits) makes the system vulnerable. Although the first estimates were too optimistic it is now believed that a fifty million dollar special purpose computer would need about two days to find a key by simply trying all possibilities (cf. [1]). There exist eavesdroppers who consider this a reasonable investment! Most experts now agree that DES will be completely insecure within 10 years. Several of the critics of the system suggested using a 128-bit key which would still be secure in a hundred years. It becomes even more remarkable when one learns that NSA restricts the export of DES chips and does not approve export licenses for cryptographic systems in which a key with more than 64 bits is used ([12]).

The second major criticism concerned the hardware. The substitution is carried out by so-called 'S-boxes'. The design of these S-boxes is kept secret, for no obvious reason, but the claim is made that the design was randomly chosen. In an analysis of DES [9] it was discovered that there is some structure in the S-boxes. Some people fear that DES also has its trap-door which enables NSA to decrypt all this allegedly safely stored information. It could be true. It became clear fairly quickly that NSA does not appreciate the revived interest in cryptography at many universities. The following incident attracted a lot of attention from the press and is still being discussed at present (cf. [7]). Shortly before Rivest was scheduled to present his results at a meeting of IEEE this organisation received a letter from an employer of NSA warning the IEEE that publication of results on cryptography might be in conflict with the International Traffic and Arms Regulation which regulates the export of weapons and sensitive equipment! The letter suggested that the authors could be prosecuted. The NSA denied involvement with the letter but a year later admiral Inman stated that open publication of research in cryptography was harmful to the security of the US. Subsequently the American Council on Education formed a study group to discuss this problem. Several of Inman's proposals were rejected but the group approved the reviewing by NSA of papers on cryptography prior to publication on the condition that compliance would be voluntary. I have been told that many authors do send their papers in for review and that two papers have actually been withdrawn by the authors at the request of NSA.

In a recent conversation with Martin Hellman he told me that he had originally strongly opposed these restrictions on academic freedom. However, during the crisis with the American hostages in Iran he had realized that it might not be such a good idea to tell everybody on earth how they can establish secure communication systems. Of course, mathematicians are aware of the fact that several other branches of science such as nuclear physics, genetics, etc. are restricted by many regulations for security and safety reasons.

Mathematicians will find it difficult to accept such restrictions. In any case it is a problem which deserves serious consideration.

REFERENCES

[1] DIFFIE, W. *Cryptographic Technology: Fifteen year forecast*, in *Advances in Cryptography* (A. Gersho, ed.) p. 84-208, Dept. of Electrical and Computer Engineering, Univ. of California, Santa Barbara.

[2] DIFFIE, W. and M.E. HELLMAN, *New directions in cryptography*, IEEE Trans. Information Theory, IT-22 (1976), 644-654.

[3] GAREY, M.R. and D.S. JOHNSON, *Computers and Intractibility, A guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.

[4] HANDELMAN, G.H. *Cryptographic research and the national security*, SIAM News 14^3(1981).

[5] HOOGENDOORN, P.J. (ed.), Cursus Cryptografie, MC-Syllabus 51, Math. Centrum, Amsterdam, 1983.

[6] KAHN, D. *The Codebreakers*, MacMillan, 1967.

[7] LANDAU, S. *Primes, Codes and the National Security Agency*, Notices AMS 30 (1983), 1-10.

[8] LEMPEL, A. *Cryptology in transition: a survey*, ACM Computing Surveys 11 (1979), 285-303.

[9] Lexar Corporation, *An evaluation of the* NBS *data encryption standard*, Unpublished report, Lexar Corporation, 11611, San Vicente Boulevard, Los Angeles, 1976.

[10] MERKLE, R. *Secure communication over insecure channels*, Communications ACM 21 (1978), 294-299.

[11] MERKLE, R. and M.E. HELLMAN, *Hiding information and signatures in trapdoor knapsacks*, IEEE Trans. Information Theory IT-24 (1978), 525-530.

[12] MORRIS, R.N.J., A. SHAMIR and A.D. WYNER, *Assessment of the National Bureau of Standards proposed federal data encryption standard*, Cryptologia 1 (1977), 281-291.

[13] POMERANCE, C. *The search for prime numbers*, Scientific American, Vol. 247, nr. 6, (December 1982), 122-131.

[14] RIVEST, R.L., A. SHAMIR and L. ADLEMAN, *On digital signatures and public key cryptosystems*, Communications ACM 21 (1978), 120-126.

[15] RUMELY, R. *Recent Advances in Primality Testing*, Notices AMS 30 (1983), 475-477.

[16] SHAMIR, A. *A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem*, preprint 1982.

[17] WILLIAMS, H.C. and B. SCHMID, *Some remarks concerning the* MIT *public-key cryptosystem*, BIT 19 (1979), 525-538.

[18] LENSTRA, H.W., Jr. *Fast prime number tests*, Nieuw Archief v. Wiskunde (4) 1 (1983), 133-144.