

## CONFIDENTIAL

Reference: HM/TECH.A/377 Date: 2 April 1998 Copy Number: 9

An Identity Based Public Key System by Clifford Cocks (HM2)

### Summary

Z TECH LIBRARY FILE # 9/60

DOCID: 3097965

In an offline public key system, in order to send encrypted data it is necessary to know the public key of the recipient. This means that directories of public keys need to be readily available. In an identity based system a user's public key is a function of his identity (for example his email address), thus avoiding the need for a separate public key directory. The possibility of such a system has been discussed for some time, but to date no satisfactory scheme has been proposed. This paper describes such a system.



## DOCID: 3097965

## -CONFIDENTIAL

HM/TECH.A/377

#### 1. Introduction

In an offline public key system, in order to send encrypted data it is necessary to know the public key of the recipient. This necessitates the holding of directories of public keys. In an identity based system a user's public key is a function of his identity (for example his email address), thus avoiding the need for a separate public key directory. The possibility of such a system has been discussed for some time (Shamir [1] is the first mention in the literature), but all proposals to date have either been computationally unattractive (e.g [2]), or have been flawed (in that users can collaborate to break the system), or they only use the identity as part of the key generating process (e.g. [3]) so that a public directory is still needed for offline communication between users.

The system described in this paper provides a workable directoryless public key system. We begin with an overview of the functionality of the system.

### 2. Overview of Functionality

The system has a universal authority U which generates a universally available public modulus M. This modulus is a product of two primes P and Q - held privately by U, where P and Q are both congruent to 3 mod 4.

Also, there will need to be a universally available secure hash function.

Then, if user A wishes to register in order to be able to receive encrypted data he presents his identity (e.g. e-mail addresss) to U. In return he will be given a private key (with properties described below).

Then, any user B wishing to send encrypted data to B will be able to do this knowing only A's public identity and the universal system parameters. There is no need for any public key directory.

#### 3. Description of the System

When A presents his identity to U, the hash function is applied to produce a value a modulo M such that the Jacobi symbol  $\left(\frac{a}{M}\right)$  is +1. This will be a public process that anyone holding the universal parameters and knowing A's identity can replicate. Essentially this will involve multiple applications of the hash function in a structured way to produce a set of candidate values for a, stopping when  $\left(\frac{a}{M}\right) = +1$ .

Thus as  $\left(\frac{a}{M}\right) = +1$ ,  $\left(\frac{a}{P}\right) = \left(\frac{a}{Q}\right)$ , and so either *a* is a square modulo both *P* and *Q*, and hence is a square modulo *M*, or else -a is a square modulo *P*, *Q* and hence *M*. The latter case arises because by construction *P* and *Q* are



### <sup>•</sup> DOCID: 3097965

## CONFIDENTIAL

HM/TECH.A/377

both congruent to 3 mod 4, and so  $(\frac{-1}{P}) = (\frac{-1}{Q}) = -1$ . Thus either a or -a will be quadratic residues modulo P and Q. Only U can calculate the square root modulo M, and he presents such a root to A. Let us call this value r. One way for U to determine a root is to calculate

$$r = a^{\frac{M+5-(P+Q)}{8}} \mod M$$

In what follows, I will assume without loss of generality that  $r^2 \equiv a \mod M$ . Users wishing to send encrypted data to A who do not know whether A receives a root of a or a root of -a will need to double up the amount of keying data they send.

Then if B wants to send a secure message to A, he first generates a cryptovariable (by whatever means he wishes) and encrypts the data using conventional encryption. He sends to A each bit of the cryptovariable in turn as follows:

Let b be a bit of the cryptovariable, coded as +1 or -1.

Then B chooses a value t at random modulo M, such that the Jacobi symbol  $\left(\frac{t}{M}\right)$  equals b.

Then he sends  $s = (t + a/t) \mod M$  to A.

A recovers the bit b as follows:

as  $s + 2r = t(1 + r/t) * (1 + r/t) \mod M$ 

it follows that the Jacobi symbol  $\left(\frac{s+2r}{M}\right) = \left(\frac{t}{M}\right) = b$ .

But A knows the value of r so he can calculate the Jacobi symbol  $\left(\frac{s+2r}{h}\right)$ , and hence recover b.

If B does not know whether a or -a is the square for which A holds the root, he will have to replicate the above, using different randomly chosen t values to send the same b bits as before, and transmitting  $s = (t - a/t) \mod M$  to A at each step.

Note The Jacobi symbol  $(\frac{x}{M})$  is the product of the two quadratic residue symbols  $(\frac{x}{P})$  and  $(\frac{x}{Q})$  (where M = PQ). Thus it is +1 if either x is a square modulo both P and Q or is a non square modulo both P and Q. A useful feature of the Jacobi symbol is that it can be calculated without knowledge of the factorisation of M. See for example [4].



### DOCID: 3097965

## -CONFIDENTIAL

HM/TECH.A/377

#### 4. Practical Aspects

Computationally, the system is not too expensive. If the cryptovariable is L bits long, then B's work is dominated by the need to compute L Jacobi Symbols and L divisions mod M. A's work mainly consists of computing L Jacobi symbols. For typical parameter values (e.g. L = 120 and M of size 1024 bits) this is likely to be less work than is needed for a single exponentiation modulo M.

The main issue regarding practicality is the bandwidth requirement, as each bit of cryptovariable requires a number of size up to M to be sent. For a 120 bit CV, and using a 1024 bit modulus M, B will need to send 15K bytes of keying material. If B does not know whether A has received the square root of a or of -a then he will have to double this. Nevertheless, for offline use such as email this may be an acceptable overhead.

#### 5. Security Analysis

Clearly, one way to break the system is to factorise M. The fact that this is a weak link means that split knowledge methods of generating M may be desirable. Our aim here is to study the security of the system on the assumption that M has not been factorised. We show that a weakness would lead to a solution of currently unsolved mathematical problems.

Suppose that there is a procedure that recovers b from s without knowing either r or the factors of M. In other words we can calculate a mapping

$$F(M, a, s) \rightarrow b = (\frac{t}{M})$$

whenever  $s = (t + a/t) \mod M$  for some t.

Then consider what the value of F could be if evaluated for an *a* where the Jacobi symbol  $\left(\frac{a}{M}\right)$  is +1, but *a* is not a square. In this case the Jacobi symbols  $\left(\frac{a}{D}\right)$  and  $\left(\frac{a}{D}\right)$  will both be -1.

Now, if t was the value used to calculate s, there will be three other values t1, t2, t3 giving the same value of s.

These are given by:

 $t1 \equiv t \mod P \qquad t1 \equiv a/t \mod Q$  $t2 \equiv a/t \mod P \qquad t2 \equiv t \mod Q$  $t3 \equiv a/t \mod P \qquad t3 \equiv a/t \mod Q$ But as  $\left(\frac{a}{P}\right) = \left(\frac{a}{Q}\right) = -1$ , then  $\left(\frac{t1}{M}\right) = \left(\frac{t2}{M}\right) = -\left(\frac{t}{M}\right) = -\left(\frac{t3}{M}\right)$ .



(b) (1)

## DOCID: 3097965

## CONFIDENTIAL

HM/TECH.A/377

So, there is no unique  $(\frac{t}{M})$  to recover, and so F cannot return  $(\frac{t}{M})$  correctly more than half the time whenever a is not a square. Hence we would have a procedure that can distinguish the two cases of  $(\frac{a}{M}) = +1$ ; that is determine whether a is a square or a non square without factoring M. To do this is currently an unsolved problem.

Now consider what F can output if M is the product of more than two primes. If F fails to output the correct value of  $\left(\frac{t}{M}\right)$  when a is a square then Fcan distinguish the cases of M having 2 prime factors or having more than two distinct factors. On the other hand, if F does output the correct value of  $\left(\frac{t}{M}\right)$ when a is a square, by the argument above, it must fail to do so if a is not a square, but  $\left(\frac{a}{M}\right) = +1$ . However, if a is randomly chosen to have  $\left(\frac{a}{M}\right) = +1$ , the probability that it is a square is approximately  $2^{1-n}$ , where n is the number of factors of M. Thus we would have a probabilistic test for the number of factors of M. Determining this without factorising M is also an unsolved problem.

### 6. References

[1] A. Shamir Identity Based Cryptosystems and Signature Schemes Advances in Cryptology - Proceedings of Crypto '84.

[2] U. Maurer and Y. Yacobi Non-Interactive Public Key Cryptography Advances in Cryptology - Proceedings of Eurocrypt '91.

[3] E. Okamoto Key Distribution Systems Based on Identification Information Advances in Cryptology - Proceedings of Crypto '87.

[4] H Cohen A Course in Computational Algebraic Number Theory Springer-Verlag graduate texts in mathematics 138, 1993

(b)(1)

# -CONFIDENTIAL