An Identity Based Encryption Scheme based on Quadratic Residues

Clifford Cocks

Communications-Electronics Security Group, PO Box 144, Cheltenham GL52 5UE

Abstract. We present a novel public key cryptosystem in which the public key of a subscriber can be chosen to be a publicly known value, such as his identity. We discuss the security of the proposed scheme, and show that this is related to the difficulty of solving the quadratic residuosity problem

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 usually 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 was first mentioned by Shamir[4], but it has proved difficult to find implementations that are both practical and secure, although recently an implementation based on elliptic curves has been proposed [3]. This paper describes an identity based cryptosystem which uses quadratic residues modulo a large composite integer.

2 Overview of Functionality

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

Also, the system will make use of a universally available secure hash function.

Then, if user Alice wishes to register in order to be able to receive encrypted data she presents her identity (e.g. e-mail addresss) to the authority. In return she will be given a private key with properties described below.

A user Bob who wishes to send encrypted data to Alice will be able to do this knowing only Alice's public identity and the universal system parameters. There is no need for a public key directory.

3 Description of the System

When Alice presents her identity to the authority, the hash function is applied to the string representing her identity to produce a value a modulo M such that the Jacobi symbol $(\frac{a}{M})$ is +1. This will be a public process that anyone holding the universal parameters and knowing Alice's identity can replicate. Typically this will involve multiple applications of the hash function in a structured way to produce a set of candidate values for a, stopping when $(\frac{a}{M}) = +1$. Note that the Jacobi symbol can be calculated without knowledge of the factorisation of M. See for example [2].

Thus as $(\frac{a}{M}) = +1$, $(\frac{a}{P}) = (\frac{a}{Q})$, 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 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 the authority can calculate the square root modulo M, and he presents such a root to Alice. Let us call this value r. One way for the authority to determine a root is to calculate

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

Such an r will satisfy either $r^2 \equiv a \bmod M$ or $r^2 \equiv -a \bmod M$ depending upon which of a or $-a$ is a square modulo M.

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

If Bob wants to send an encrypted message to Alice, he first generates a transport key and uses it to encrypt the data using symmetric encryption. He sends to Alice each bit of the transport key in turn as follows:

Let x be a single bit of the transport key, coded as $+1$ or -1.
Then Bob chooses a value t at random modulo M, such that the Jacobi symbol $(\frac{t}{M})$ equals x.
Then he sends $s = (t + a/t) \bmod M$ to Alice.

Alice recovers the bit x as follows:
as $s + 2r = t(1 + r/t) * (1 + r/t) \bmod M$
it follows that the Jacobi symbol $(\frac{s+2r}{M}) = (\frac{t}{M}) = x$.
But Alice knows the value of r so she can calculate the Jacobi symbol $(\frac{s+2r}{M})$, and hence recover x.

If Bob does not know which of a or $-a$ is the square for which Alice holds the root, he will have to replicate the above, using different randomly chosen t values to send the same x bits as before, and transmitting $s = (t - a/t) \bmod M$ to Alice at each step. This doubles the amount of keying data that Bob sends. It would be useful to find a way to avoid having to send this extra information, but at present this is an unsolved problem.

4 Practical Aspects

Computationally, the system is not too expensive. If the transport key is L bits long, then Bob's work is dominated by the need to compute L Jacobi symbols and L divisions mod M. Alice's work mainly consists of computing L Jacobi symbols. For typical parameter values (e.g. $L = 128$ and M of size 1024 bits) and depending upon the implementation this is likely to be no more work than is needed for a single exponentiation modulo M.

The main issue regarding practicality is the bandwidth requirement, as each bit of the transport key requires a number of size up to M to be sent. For a

128 bit transport key, and using a 1024 bit modulus M, Bob will need to send 16K bytes of keying material. If Bob does not know whether Alice 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 will often 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 shared knowledge methods of generating M (see [1] for example) and the use of multiple authorities will be desirable. With shared generation of M it is also feasible to generate the exponent $\frac{M+5-(P+Q)}{8}$ used to compute square roots in a shared fashion, so that no master secret ever needs to exist in a single location.

We study the security of the system on the assumption that M has not been factorised. We consider first a passive attack against the generation of each bit of transport key and show that a weakness would lead to a solution of the quadratic residuosity problem.

Suppose that there is a procedure that recovers x from s without knowing either r or the factors of M. We also assume a constraint on the hash function, that the recovery procedure takes as input the hashed identity a, and can not make separate use of the input to the hash. This excludes obviously weak hash functions, such as one whose final step consists of squaring modulo M. Under this hypothesis the breaking process computes a mapping

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

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

Then consider what the value of F could be if evaluated for an a where the Jacobi symbol $(\frac{a}{M})$ is +1, but a is not a square. In this case the Jacobi symbols $(\frac{a}{P})$ and $(\frac{a}{Q})$ 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:

$$\begin{array}{ll} t1 \equiv t \bmod P & t1 \equiv a/t \bmod Q \\ t2 \equiv a/t \bmod P & t2 \equiv t \bmod Q \\ t3 \equiv a/t \bmod P & t3 \equiv a/t \bmod Q \end{array}$$

But as $(\frac{a}{P}) = (\frac{a}{Q}) = -1$, then $(\frac{t1}{M}) = (\frac{t2}{M}) = -(\frac{t}{M}) = -(\frac{t3}{M})$.

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. This is the quadratic residuosity problem which is currently unsolved, and is a problem on which a number of other public key systems are based.

Of course, an attacker will in practice be presented with a set of many such terms $(t + a/t) \bmod M$ and possibly also $(t - a/t) \bmod M$ for different values of t. It is desirable that the values of t used are independent and randomly distributed over the set of values consistent with the desired key value. For if successive values of t are related in a systematic way this opens up the possibility of an attack against the set of transmitted values.

The scheme as described so far is vulnerable to an adaptive chosen ciphertext attack. Because the transport key is established one bit at a time, an attacker could take a target transmission and modify the component corresponding to just one bit of transport key at a time, changing it to produce a transport key value known to the attacker. By observing the decrypt to see whether this changes the transport key the attacker could recover the transport key a bit at a time.

I sketch here an outline of how one might block such attacks. The approach is to add redundancy to the transport key establishment data so that only a small proportion of randomly chosen messages will decrypt in a valid way, and arrange that if the recipient is presented with an invalid message then the only output will be the information that the message is invalid. This should be done in a way that prevents an attacker devising challenges which may be of use in an attack. For the system described here we propose sending, suitably encrypted, data that will allow the t values to be reconstructed and then checked by the recipient, along with a cryptographic hash of those t values. This string would be produced separately for the two cases of square root (a and $-a$ respectively) that may be held by the recipient.

References

1. C Cocks *Split Generation of RSA Parameters with Multiple Participants* Proceedings of 6th IMA conference on Cryptography and Coding, Springer LNCS 1355.
2. H Cohen *A Course in Computational Algebraic Number Theory* Springer-Verlag graduate texts in mathematics 138, 1993
3. D Boneh, M Franklin *Identity-Based Encryption from the Weil Pairing* Advances in Cryptology - Crypto2001, Springer LNCS 2139
4. A. Shamir *Identity Based Cryptosystems and Signature Schemes* Advances in Cryptology - Proceedings of Crypto '84.