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