Potential of eth Residuosity in Identity Based Encryption

Rio LaVigne, Manuel Sabin
Advised by Dan Boneh

Introduction

Unlike other forms of public key encryption, where someone’s public key must be authenticated or used anonymously, an Identity Based Encryption scheme (IBE) can take an arbitrary string, convert it into a usable public key. Now, just by knowing the identity of the person you want to send a message to, you can derive their public key.

The first working IBE was the Boneh-Franklin scheme created in 2001 [1], using bilinear pairings. Although computationally inefficient, it relies on newer assumptions. Shortly after this scheme, Clifford Cocks’s used the much older Quadratic Residuosity assumption [3]. However, this scheme is not very efficient. This project looked at a generalization of the Cocks’s scheme and attempted to make it more space-efficient.

Definitions

ER(N) is the set of eth residues modulo N.

PR(N) is the set of elements such that the eth power residue symbols 1 are modulo N. That is, \( \left( \frac{1}{x} \right)_e = 1 \).

\( p \) and \( q \) are prime integers.

\( Z_p \) represents integers mod N.

\( R/J \) is a ring R mod an ideal J.

\( x \) represents the ideal generated by x.

\( \zeta \) represents a primitive eth root of unity. When \( e \) is implied by context, it is left off.

eth Residuosity Assumption (ER)

We will assume that knowing the factorization of N, determining if an element a came from ER(N) or from PR(N) \( \setminus \) ER(N) is computationally hard without knowing the factorization of N.

The Problem

The current generalization of the Cocks’s IBE scheme is horrendously space-inefficient. In order to send a function of degree e - 1, it relies on newer assumptions.

Shortly after this scheme, Clifford Cocks’s used the much older Quadratic Residuosity assumption [3]. However, this scheme is not very efficient. This project looked at a generalization of the Cocks’s scheme and attempted to make it more space-efficient.

Approaches

The general scheme scales very poorly, needing to send a polynomial of degree e - 1 for each message. We looked for ways to send a function of lower degree or compress the function of higher degree.

1. Our main approach: send a function of smaller degree.
2. Compress the representation of the higher degree function using families of functions with the same ratios or ideals

References