Abstract—In 1985, Victor Miller and Neil Koblitz, independently, proposed the use of elliptic curves in public key cryptosystems, known as elliptic curve cryptography (ECC). The security of ECC is based on the intractability of the elliptic curve discrete logarithm problems (ECDLP). The key advantage of ECC over other established algorithms such as RSA or DSA is that there is no sub-logarithmic algorithm for solving the ECDLP. The lack of a sub-logarithmic algorithm results in substantially smaller key lengths, with the same security, as RSA or DSA. This makes ECC highly attractive when implemented on devices with limited resources. Elliptic curves have the potential, due to the lack of a credible attack, of replacing RSA as the algorithm of choice in public key cryptosystems.

I propose to build a system that employs elliptic curves as the primary encryption algorithm. The elliptic curves will be defined over the fields $F_{2^m}$ and $F_p$ with an underlying arithmetic based on polynomial and optimal normal representation.

Index Terms—ECC, RSA, DSA, Encrypt, ECDLP, polynomial, optimal normal basis.

I. INTRODUCTION

Cryptography has in recent times become the watchword when talking about protecting data on the Internet. Media reports continue to carry reports of hacker activity and the resultant espionage. This has directly affected consumer confidence in activities such as e-commerce. Most consumers are wary of providing confidential information such as credit card numbers to web pages because of the security risks. Numerous software tools exist that provide cryptographic support to Internet activities, such as SSL, the de facto standard in web-page encryption. SSL and most other types of security packages use RSA as their primary algorithm.

Elliptic curves have interested mathematicians for the last 150 years. This has led to a very complex and deep theory. In 1985, Victor Miller and Neil Koblitz independently proposed the use of elliptic curves in public key cryptography. As mentioned previously, the security of elliptic curve cryptography is based on the ECDLP [2]. The security of the RSA algorithm is based on the integer factorization problem [2] while DSA is based on the discrete logarithm problem [2]. It is considered by many experts that of the three hard mathematical problems, the ECDLP is the most difficult to solve.

One of the problems associated with developing an ECC system is that the underlying mathematics of elliptic curves is very difficult to understand if one is an “amateur” mathematician. The research aim is therefore to design an e-commerce system protected by a system based on SSL. The core algorithm will be based on the theory of elliptic curves. This cryptographic system will allow encryption (ECAES), signature verification and key transport (ECDH, ECDSA).

II. TECHNICAL OVERVIEW

Fig1: An elliptic curve over the field $F_q$

Elliptic curves are defined over finite fields $F_q$ which are required for cryptographic implementations. Finite fields always consist of a finite number of elements. For cryptographic purposes, the finite field $F_p$ (called a prime field) and the binary finite field $F_{2^m}$ will be considered.

The Field $F_p$

Let $p$ be a prime number, then the field $F_p$ consists of a set of integers $\{0, 1, 2, 3, \ldots, p-1\}$. The following arithmetic is allowed over this field. [1], [5]

- Addition: if $a$, $b$ are elements of $F_p$ then $a + b = c$, where $c$ is the remainder when $a + b$ is divided by the prime $p$. 

Multiplication: if \( a, b \) are elements of \( F_p \) then \( a \times b = c \), where \( c \) is the remainder when \( a \times b \) is divided by \( p \).

The above operations all end in reduction modulo \( p \).

**The Field \( F_{2^m} \)**

The representation of this field is more complex than the field \( F_p \). One of the major advantages of this field is that the field \( F_{2^m} \) uses binary notation and is hence much more efficient than the field \( F_p \). The underlying representation can be either of the following bases. Note: these bases are not interchangeable.

- **Polynomial Basis:**
- **Optimal Normal Basis:**

**A. Polynomial Basis**

Polynomial basis allows the following operations, namely

- Addition: \( a + b = c \), where \( c_i = (a_i + b_i) \mod 2 \), i.e. bit wise exclusive-or.
- Multiplication: \( a \times b = c \), where \( c \) is the product of polynomial division.

**B. Optimal Normal Basis**

Optimal normal basis allows the following operations, namely:

- Addition: \( a + b = c \), where \( c_i = (a_i + b_i) \mod 2 \), i.e. field addition is performed bit wise.
- Squaring: A rotation of the vector representation.
- Multiplication: \( a \times b = c \), where indices are reduced modulo \( m \).

**Elliptic Curves defined over Prime Fields**

Elliptic curves over \( F_p \) are defined by equations of the form \( y^2 \mod p = x^3 + ax + b \mod p \). The variables \( a \) and \( b \) are chosen so that \( a, b \) are elements of \( F_p \). To ensure that the curve together with the field forms a group\(^1\), it is important that \( x^3 + ax + b \) has no repeating factors or that \( 4a^3 + 27b^2 \) is not equal to zero.

**A. Point Addition over \( E(F_p) \)**

One of the operations allowed on an elliptic curve defined over a prime field is adding two points to get a third. The third point is guaranteed to be on the curve.

\[ P + Q = R^2 \]

\(^1\) A group consists of all the points on the curve as well as the point of infinity.

**B. Doubling a Point**

Another property of elliptic curves defined over prime fields is doubling a point. A specific algorithm exists to double a point. Let \( P = (x_1, y_1) \) then

\[ 2P = R, \]

1. \( s = (3x_1^2 + a)(2y_1) \mod p \)
2. \( x_R = s^2 - 2x_1 \mod p \)
3. \( y_R = -y_1 + s(x_1 - x_R) \mod p \)

**Elliptic Curves defined over \( F_{2^m} \)**

Elliptic curves defined over finite fields of the form \( F_{2^m} \) can be expressed as equations in the form

\[ y^2 + xy = x^3 + ax^2 + b, \]

together with a special point called the point of infinity\(^1\).

**III. RESEARCH METHODOLOGY**

The cryptographic system will be based on elliptic curves. The elliptic curves will be defined on finite fields over the fields \( F_p \) and \( F_{2^m} \).

The system will entail the following:

- Choice of underlying finite field (i.e. either \( F_p \) or \( F_{2^m} \)).
- Implementation of the underlying elliptic curve arithmetic.
- Generation of strong primes using accepted algorithms \(^8\).
- The generation of random elliptic curves over finite fields.
- The encryption of data “onto” the curve.
- Generation of public and private keys.
- Key transport via ECDH \(^6\) and ECDSA
- Data encryption using ECAES.
- Generation of signatures to ensure data integrity.

The development of an elliptic curve cryptosystem is quite complex and requires one to make the following choices \(^2\), namely:

- Choosing the type of finite field.
- Choosing the various algorithms for implementing the finite field arithmetic.
- Choosing a particular elliptic curve.
- Choosing algorithms for implementing the elliptic curve group arithmetic.
- Choosing an elliptic curve protocol.

The system will be used to protect a web-based e-commerce application. All traffic to and from the web page will be
encrypted with ECC. The system will perform the following functions:

- Generation of a session elliptic curve (either $F_p$ or $F_{2^m}$).
- Generation of public and private keys (points on the curve).
- Transfer of data between server and user (based on ECDH or DSA).

One of the key tests will be to deploy a packet sniffer on the network. The packet sniffer will be used to view the contents of each packet as it travels across the network. The test will also ascertain whether one can work out the private keys based on the contents of the packets.

IV. RESULTS

An elliptic curve cryptosystem was built based on the field $F_{2^m}$. The underlying arithmetic is based on optimal normal basis.

The elliptic curve system uses the Diffie-Hellman [7] key exchange protocol. This protocol uses public and private keys to encrypt data during transportation.

At this stage the public keys are transported across a single socket on a single computer.

The system uses two copies of the elliptic curve software (a client and a server) and works as follows:

- The client connects to the server across a socket.
- The client downloads a copy of the elliptic curve parameters, creates a session curve and chooses a base point on the curve.
- The client then chooses a random point as the public key and keeps the base point secret.
- This random point is sent across the network to the server.
- The server sends a test string of digits that has been added onto the curve.
- The client receives the encrypted string and decrypts the data by using its private key.

V. FUTURE WORK

Work still to be completed include the following:

- Extending the system to include the polynomial basis.
- Implementing elliptic curves over the field $F_p$.
- Implementing the e-commerce application that will be used to test the system.
- Testing the system with a packet sniffer.

CONCLUSIONS

Elliptic curve cryptography, due to the lack of a sub-exponential attack, has the ability to replace RSA as the de facto cryptographic algorithm. While in certain circumstances the algorithm is vulnerable [1], some simple tests and design choices makes elliptic curve cryptography extraordinarily hard to break.

ACKNOWLEDGMENT

This research has been funded by the Telkom-Cisco Centre of Excellence for IP and Internet Computing and the South African National Research Foundation.

BIOGRAPHIES

Ashraf Salie – Ashraf Salie is a Masters candidate in Computer Science. His research interest is the fields of Internet security and cryptography.

Christian Omlin – Christian Omlin is Professor of Computer Science at the University of the Western Cape. His research interest includes artificial intelligence.

REFERENCES


3 RSA and DSA has sub-exponential attacks.