

## An Elliptic Curve Cryptographic System Design Architecture with Application to Distributed Simulation

P.H. Roberts, R.N. Zobel<sup>1</sup>

*Department of Computer Science, University of Manchester, Oxford Road  
Manchester M13 9PL, United Kingdom.*

**Abstract** - Distributed simulation, outside of the military area, necessarily operates over the internet, which implies the risk of many forms of attack. Current security systems offer limited protection because of the cost and complexity of using sufficient key length in existing public key encryption schemes. The use of the Discrete Logarithm Problem over elliptic curves defined over finite fields as a basis for trap-door based public key encryption (ECC) appears to offer improved performance with lower cost in terms of processor speed, memory requirement and processing time. This paper provides an outline of ECC and the complexities of a practical implementation of the technology. Some issues regarding choice of EC parameters, security, interoperability and performance are discussed. A proposal is made for a tool set to enable development of a broad range of elliptic function based methods, by providing the top level of required modules. This enables further development of particular encryption schemes in a structured way to meet the particular needs of the cryptographic systems designer. Such cryptographic systems may be considered suitable for supporting distributed interactive simulation, with its stringent timing requirements and particular security problems, and with special reference to mobile systems.

**Keywords** - *Cryptography, Elliptic Curve, Distributed Simulation, ECC, System Architecture, Authentication*

<sup>1</sup> *Assoc. Prof. Dr. Richard Zobel is now retired, but works periodically at the Prince of Songkla University, Phuket, Thailand.*

### I. INTRODUCTION

Increasingly, distributed simulation is required for a variety of reasons, such as models running under differing operating systems, clock speeds or simulation environments, or for reasons of commercial, government or military secrecy. For cost reasons, many of these systems now have to use the internet to carry their intercommunication traffic. Consequently, the risk of lost or modified data, loss of secrecy and interference with simulation studies, makes it imperative to use some form of encryption. Further, it is necessary to provide adequate authentication facilities for all intended participants in a shared distributed simulation environment.

#### A. Attacks

There are many forms of attack. Passive attacks concern listening, observing and collecting information. Active attacks include masquerade, replay, modification, and denial of service. Of these the last is the most dangerous, since the others can be defended against by using authentication and encryption. Denial of service and worse still, distributed denial of service, can be targeted or undirected random attacks, but are less likely to be targeted if it is difficult to identify who are the participants in a distributed simulation exercise and where on the network they are.

#### B. Security Services

There are several important aspects of security which directly impact on the use of authentication and encryption for distributed simulation. They relate to maintaining privacy, strictly protecting data and messages, and system integrity during the set-up, simulation and post simulation analysis. Further, sophisticated authentication procedures ensure that only bona fide participants are permitted to join a federation, whether active in simulations or just as observers.

**B1. Confidentiality:** This concerns providing protection against passive attacks, such as acquisition of data by copying and traffic analysis for operational analysis. Encryption is often used to provide protection against such attacks.

**B2. Integrity:** Maintaining system integrity is, at a lower level, concerned with ensuring that the contents of a message have not been changed. Full integrity concerns the entire message stream making up a communication, ensuring that no messages or parts of messages have disappeared, have been replayed or that no additional messages or parts thereof have been inserted by active attacks.

**B3. Authentication:** Masquerade attacks occur when unauthorized participants assume the identity of those who are authorized. Authentication concerns ensuring that, at all

times, the authorized principals are in fact who they claim to be.

**B4. Non-Repudiation:** This is not necessarily of particular interest to simulationists, but ensures that once a party has sent a communication, they are not able to deny having sent it, and that the receiving party can prove that the original party did indeed send the message.

**B5. Symmetric/Asymmetric Key Encryption:** Discussed in detail later, this concerns the use of symmetric, shared encryption/decryption key, schemes with their associated problems of key distribution and management, and asymmetric, public key, systems.

**B6. Current limitations and Susceptibility:** The use of authentication and encryption for distributed simulation, and other related real-time activities, is limited by the additional time delay imposed by message construction, interchange and message encryption/decryption at each end of each interaction pair. With the use of real equipment and personnel the issue of mobile simulation arises and brings up the general issue of the use of mobile computing devices.

First there is the issue of the physical security of small portable digital devices such as mobile phones, PDAs and smart cards. They are easily lost or stolen.

Then there are the limited resources of such devices which limit the complexity and degree of security currently possible. For example processor power and memory size constrain the complexity of encryption/decryption algorithms. Additionally, there is limited bandwidth available in wireless networks resulting in transmission speed restrictions. Finally, there is the basic problem of the insecure nature of the radio medium.

## II. ENCRYPTION

Many methods have been used over the centuries with varying degrees of success. However, with the advent of computers, life has become easier for the attacker. Cryptanalysis has become a sophisticated topic, available to security professionals, criminals and hackers alike. Consequently, existing systems are under threat of serious attack. Unconditional security is not a reality. Realistic conditions for computational security might be that the cost of breaking the cipher is more than the value of the encrypted data, and the time to break the cipher is longer than the useful lifetime of the encrypted data [Stallings W. 1995]. The latter condition can, in principle, be obviated by the use of time stamps. The first condition is less easy to achieve, because of novel backdoor approaches and clever mathematicians.

### A. Classical or Symmetric Encryption

This usually involves three players. Two are the communicants (the principals) and the third is the attacker. The communicants use a publicly known encryption scheme and share a secret key. The important concepts are that the same secret key is used for both encryption and decryption, and usually, the decryption algorithm is simply the inverse of the encryption algorithm. Although potentially very efficient, there are three problems with such schemes. The first is key distribution, which must be in itself secure, the second is key management, where the number of keys required in a system with a large number of principals does not scale well, as shown in figure 1.

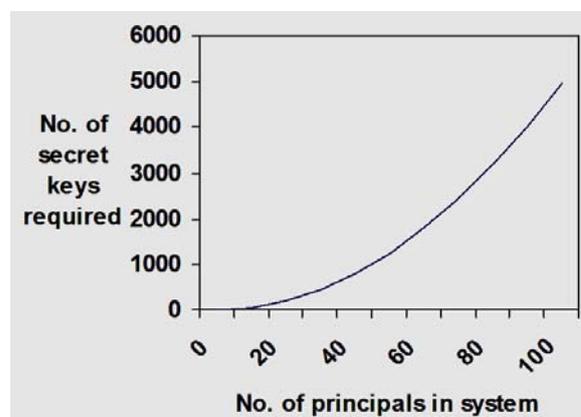


Figure 1. Number of keys required for symmetric encryption system

This can be a major problem for distributed simulation with a large number of players. Thirdly, symmetric key cryptographic systems suffer from a lack of provision for strong authentication, digital signatures and non-repudiation services. These problems can be solved through the use of Public Key encryption schemes, for a full discussion see [Stalling W. 1999].

### B. Public Key or Asymmetric Encryption

Public Key Cryptosystems employ asymmetric key pairs, where each user has an individual key pair consisting of a publicly available encryption key, and a corresponding private decryption key. It should be computationally infeasible to compute the private key from knowledge of the corresponding public key and encryption algorithm, or from examples of encrypted and decrypted message pairs. The consequence of this is the need to find so called trap-door one-way functions (TOFs) which fulfill these criteria. A trap door is easy to open from the inside, but difficult from the outside without the key. The RSA scheme [Rivest. et al, 1978] uses the product of large prime integers as the basis of a family of TOFs. However to provide adequate security it is necessary to use a key with at least 1024 bits, and the encryption/decryption operations are orders of magnitude

slower than conventional symmetric cryptosystems. Thus its practical use in distributed simulation is limited. It has been suggested that ECC is significantly more efficient than RSA, due to shorter key lengths whilst providing similar levels of security, resulting in smaller memory, CPU and bandwidth requirements and faster encryption/decryption algorithm execution. This led to the investigation of ECC as a practical alternative to RSA.

### III. ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

ECC schemes are based on the scalar multiplication of elliptic curve points and the computational intractability of the inverse operation, the Elliptic Curve Discrete Logarithm Problem (ECDLP).

#### A. Elliptic Curves

An elliptic curve  $E(F)$ , is the set of solutions, or points, which satisfy a Weierstrass equation, given by:

$$y^2z + a_1xyz + a_2yz^2 = x^3 + a_3x^2z + a_4xz^2 + a_5z^3$$

Where the  $a_i$  and the coordinates of each point are elements in a field  $F$ .

#### B. Finite Fields

For a full introduction to the theory of fields and finite fields, see [Koblitz, 1987]. Two types of field are suitable for ECC, namely, large prime characteristic finite fields and characteristic two finite extension fields. In both cases the field chosen can be defined by its order, denoted  $q$ , which is the number of elements in the field. Additionally the Weierstrass equation of elliptic curves defined over such fields can be simplified and the chosen curve defined by two parameters, commonly denoted  $a$  and  $b$ .

#### C. Underlying Mathematical Issues

There have been a number of different ECC cryptographic schemes proposed in the literature, most of which are based on the scalar multiplication, or repeated addition, of EC points. The addition of two EC points is illustrated in figure 2 below.

The scalar multiplication operation dominates the actual execution timing of an ECC scheme and thus the efficient implementation of the operation is crucial. The actual mathematics depends on the curve and underlying field chosen, however four basic operations are needed in the underlying finite field,

- Addition of two field elements
- Subtraction of field elements
- Multiplication of two field elements
- Multiplicative inversion of a field element

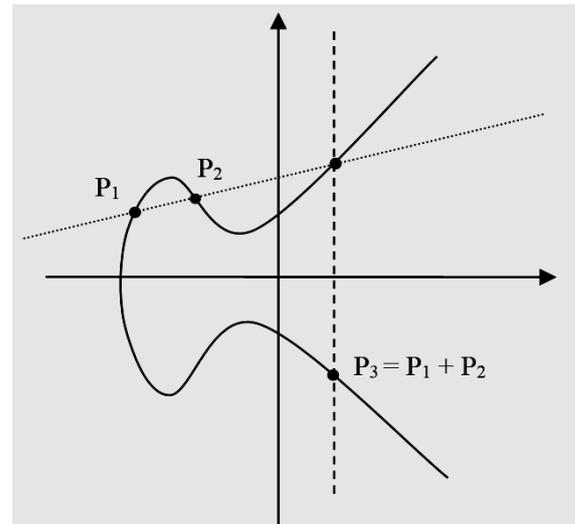


Figure 2. Illustration of adding two EC Points.

EC point compression techniques also require square root computation in the field. These operations require integer arithmetic, which may have operand lengths considerably larger than the computer word length. The choice of underlying mathematical algorithms thus depends first on the chosen curve and type of underlying finite field and then can often be optimized depending on the computing environment. Clearly in a distributed simulation environment the devices involved may have widely differing computing capabilities and architectures. Thus for optimal performance the underlying mathematical algorithms may need to be configured on a per device level.

### IV. ECC SYSTEM IMPLEMENTATION ISSUES

There are many issues to be discussed between the mathematical basis for an ECC scheme and a practical working ECC system. These are discussed in detail in the MSc thesis [Roberts P.H., 2004]. Below is not an exhaustive but indicative list of relevant issues. It will be seen from this that many decisions have to be made at several levels.

#### A. EC Parameters

ECC schemes operate on a sub-group of EC points on an elliptic curve. The group of points to be used can be specified by an EC parameter set, which can be defined as a 7-tuple,

$$(a, b, q, G, n, h, Fr)$$

Where,  $a$  and  $b$  define the elliptic curve,  $q$  identifies the underlying finite field,  $G$  is a base point which generates the chosen sub-group of points on the curve,  $n$  gives the number of points in the sub-group and  $h$  gives the total number of

points on the curve. Fr can be used to give an indication of the representation to use for the underlying field elements. This is important for characteristic two extension fields, where a number of different basis for field element representation are possible.

*A1. Parameter Set Selection:* Parameter sets may chosen from a public list of secure sets, however these may have been investigated by hackers. Alternatively, you may choose you own set of parameters. This involves selection of a finite field and generation of a curve, followed by selection of a suitable sub-group and optionally a field representation. The number of points on the curve can be checked as suitably secure using Schoof's algorithm [Schoof, 1985]. A further technique is to use a random number generator to select a curve, over a finite field, using an arbitrary seed string, which is also then included in the parameter set for later verification. This attempts to avoid the possibility of selecting a curve with a hidden cryptographic weakness.

*B. Security*

The level of security offered by an ECC scheme is largely determined by the difficulty of solving the ECDLP over the group of EC points used with the scheme. This should be a prime-order cyclic sub-group of points on a suitably secure elliptic curve. It is essential that only groups where the ECDLP is computationally infeasible be used. Figure 3 [ANSI X9.62, 2001], reproduced below, shows that reasonable sizes of n give very long time periods for solving the ECDLP, using the Rho algorithm [Pollard, 1987].

Size of n (bits)	$\sqrt{(\pi n/4)}$	MIPS years
160	$2^{80}$	$8.5 \cdot 10^{11}$
186	$2^{93}$	$7.0 \cdot 10^{15}$
234	$2^{117}$	$1.2 \cdot 10^{23}$
354	$2^{177}$	$1.3 \cdot 10^{41}$
426	$2^{213}$	$9.2 \cdot 10^{51}$

Figure 3. Computing time estimates for solving the ECDLP for various values of n

There are a number of special classes of curves where algorithms of sub-exponential complexity for solving the ECDLP are known, these curves should be avoided.

It should be noted that it is the order of the base point of a parameter set n, which determines the difficulty of solving the ECDLP and the order of the underlying field q, that determines the size of the keys generated. From this it should be clear that in a security system requiring more than one level of security a separate parameter set is needed for each security level and that each user will require a separate key-pair for each parameter set in use in the system.

*C. Interoperability*

To achieve interoperability at any level, i.e. system wide or inter-system, it is necessary that all participating users of a scheme have key-pairs based on a shared EC parameter set. In practice there are two additional requirements,

- Use of standardized descriptions of cryptographic primitives
- Use of standardized versions of the common ECC schemes and protocols

For performance and security reasons, the possibilities for interoperation may be considerably reduced.

*D. Performance*

Parameter set selection has a large bearing on performance, particularly with respect to the possible implementation of the EC mathematics. Optimization by efficient coding in assembler or with hardware implementation may well be necessary for real-time applications. Network performance depends on common standards for formatting and encoding, plus of bandwidth limitations. Ultimately, there will always be a difficult three sided compromise between interoperability, performance and security, which is application dependent.

*E. Implementation*

Optimal performance can be obtained by designing the complete ECC system tailored to the individual needs of the application, but this is usually expensive and requires expert knowledge. A second alternative is a two level tailored approach, encompassing first the perspective of the application and then second the perspective of the device/platform on which the ECC services will run:

*E1. Application Level Issues:* These are most naturally considered from an application level perspective:

- Level of security
- Conformance to standards
- Selection of cryptographic schemes
- Formats for network transfer of keys and other cryptographic primitives

*E2. Device Level Issue:* Device level issues concern the selection and optimization of the available underlying mathematical algorithms, based on the parameter set chosen at the application level, to give the best performance for the computing resources available on the device which involves:

Selection of internal field element representation and associated mathematical algorithms.

Selection of internal point representation and selection and optimization of point addition algorithms.

Selection and optimization of a scalar multiplication algorithm taking side channel attacks into account.

Utilization of any special purpose field or integer arithmetic hardware available on a specific device.

*E3. Toolkit Approach:* Such tailored approaches still imply that a new ECC system must be built from scratch for

each new application that wishes to use ECC. This is inevitably expensive in terms of application development. A third approach is therefore to provide a toolkit of re-usable ECC algorithm components, which can be plugged together in a framework to build a customized ECC solution. Figure 4 shows the high level architecture of such a two layered toolkit approach.

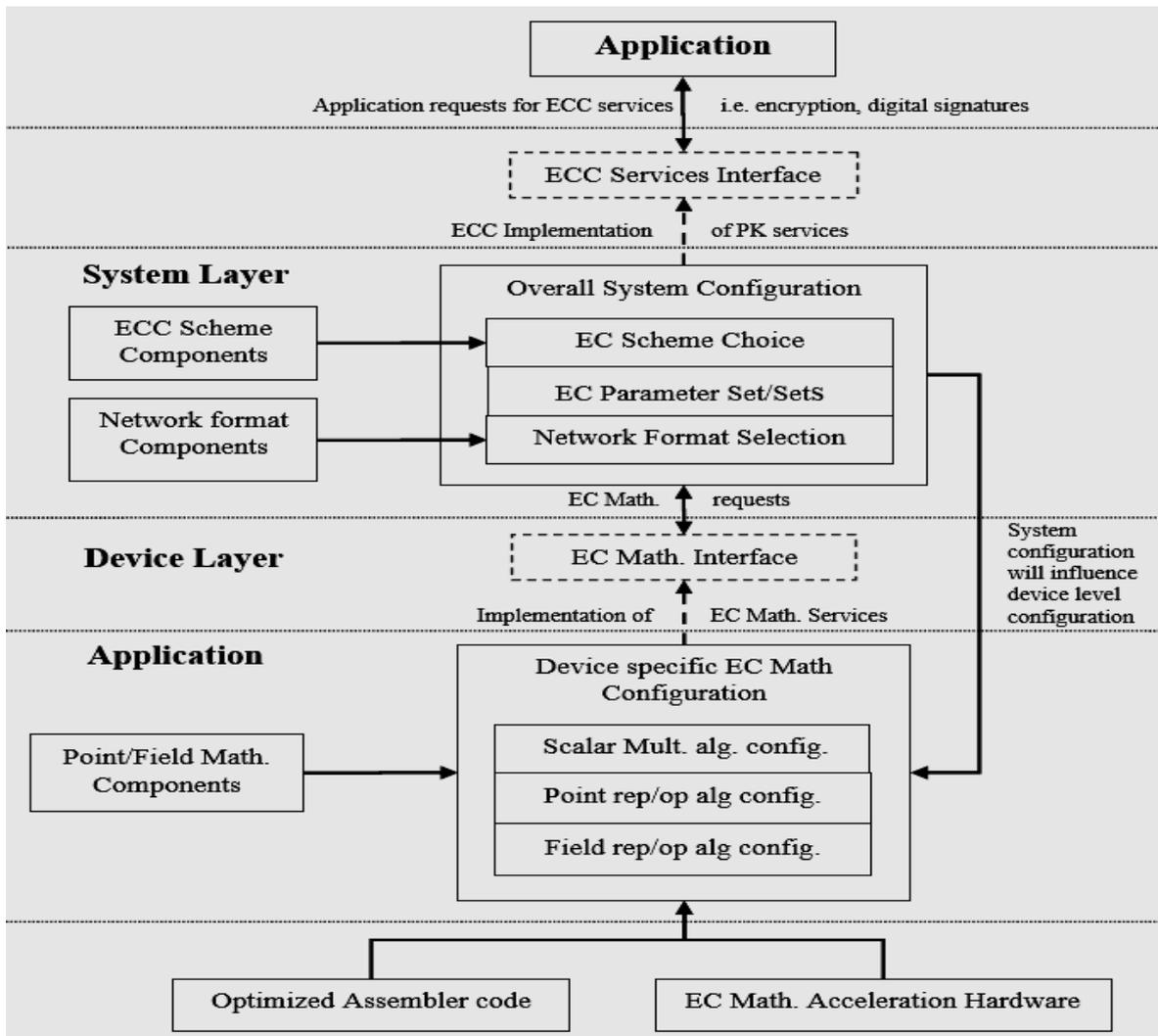


Figure 4. A Two Layered Toolkit Approach

As can be seen, this involves two important interfaces:

- An ECC Services Interface between the System Layer and the Application.
- An EC Mathematics Interface between the Device Layer and the System Layer.

The ECC Services interface needs to allow the application to request ECC versions of the three common public key cryptographic services:

- Computation of digital signatures
- Encryption/decryption of data
- Computation of secret key values through key exchange schemes.

This interface separates the implementation of the ECC system from the cryptographic services provided by the configured ECC system. The EC Mathematics interface separates the configured ECC system from the

implementation of the underlying EC Mathematics required by the ECC schemes. This enables application level issues, which affect the entire system, to be resolved in the system layer. Additionally this second interface abstracts the EC mathematics services needed by the ECC schemes from the underlying device/platform specific implementation of the mathematics.

#### V. CONCLUSIONS AND IMPLICATIONS FOR SIMULATION

It is clear that elliptic curve cryptography is complex and there are a number of practical issues to be resolved when integrating the technology into a security system. For successful use of distributed simulation, whether for discrete event, continuous or mixed, and whether for large numbers of participants (federates) or smaller federations, or whether for PCs or supercomputers, there is a real need for adequate protection against attack from any quarter. One must consider the performance aspects in respect of the time penalty for the use of authentication and encryption/decryption in real-time applications. This paper has provided a brief view into the complex topic of elliptic curve cryptography. It has enormous potential for industrial and commercial use both inside and outside of computer

simulation, particularly where cost dictates the use of the internet. However, it is also clear that there remain problems of agreed standards, acceptability, and of memory and processor requirements and limitations, particularly for the increasingly important area of mobile applications.

#### REFERENCES

- [1] ANSI X9.63, 2001. "Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography".
- [2] Koblitz, N. 1987. "A Course in Number Theory and Cryptography". Springer-Verlag. ISBN: 0-387-96576-9
- [3] Pollard, J. 1978. "Monte Carlo Methods for Index Computation Mod- p". Mathematics of Computation, vol. 32, p918-924.
- [4] Rivest, R; Shamir, A and Adleman, L. 1978. "A Method for obtaining Digital Signatures and Public Key Cryptosystems". Communications of the ACM, Feb.
- [5] Roberts, P.H. 2004. MSc Thesis, University of Manchester, U.K.
- [6] Schoof, R. 1985. "Elliptic Curves over Finite Fields and the Computation of Square Roots Mod p". Mathematics of Computation, vol. 44, p483-494.
- [7] Stallings, W. 1995. Network and Internetwork Security Principals and Practice. Prentice Hall. ISBN 0-13-180050-7.
- [8] Stallings, W. 1999. Cryptography and Network Security Principals and Practise (2nd Ed.) Prentice Hall. ISBN: 0-13-869017-0.
- [9] Stinson, D. 2002. Cryptography Theory and Practice (2nd Ed). Chapman and Hall/CRC. ISBN 1-58488-206.