RSA Home
Search
Sitemap

About RSA
Careers at RSA
Contact RSA
Developers Corner
Events
**FREE** Software & Publications
Partner Products
Pressbox
Pricing
RSA Art Gallery
RSA Products
Standards

RSA Japan
RSA Laboratories
Security Dynamics

# Elliptic Curve Cryptosystems

*M.J.B. Robshaw, Ph.D. and Yiqun Lisa Yin, Ph.D.*

*An RSA Laboratories Technical Note*
*Revised June 27, 1997*

**Abstract.** *Elliptic curve cryptosystems appear to offer new opportunities for public-key cryptography. In this note we provide a high-level comparison of the RSA public-key cryptosystem and proposals for public-key cryptography based on elliptic curves.*

## 1. Introduction and History

The mathematical idea fundamental to public-key cryptography is that of a hard problem and from such problems, mechanisms for public-key key exchange might be constructed [DH76]. If an additional technical requirement (a *trapdoor*) can be designed then the hard problem might possibly be used to construct a public-key encryption or a digital signature algorithm [DH76].

While the 20-year history of public-key cryptography has seen a diverse range of proposals for candidate hard problems only two have truly stood the test of time. These problems are known as the *discrete logarithm problem over a finite field* and *integer factorization* [Sim92].

During the mid-1980's various researchers observed [Mil86, Kob87] that another source for hard problems might be discovered by looking at *elliptic curves*. Elliptic curves are rich mathematical structures which have shown themselves to be remarkably useful in a range of applications including primality testing and integer factorization [Len87, Men93]. One potential use of elliptic curves is in the definition of public-key cryptosystems that are close analogs of existing schemes. In this way, variants of existing schemes can be devised that rely for their security on a different underlying hard problem.

The aim of this note is to provide an overview of the different tradeoffs involved in choosing between cryptosystems based on elliptic curves and RSA [RSA78]. We will not, however, be providing any mathematical details of either cryptosystem, nor will we offer details of the calculations performed when making our comparison between the two systems.

## 2. Elliptic Curve Cryptosystems

The proposed elliptic curve cryptosystems are analogs of existing schemes. It is possible to define elliptic curve analogs of the RSA cryptosystem [Dem94, KMOV92] and it is possible to define analogs of public-key cryptosystems that are based on the discrete logarithm problem (such as ElGamal encryption [ElG85] and the DSA [NIST94] for instance). The case of analogs to the discrete logarithm problem can be divided into two classes. In the first class the finite field is said to have *odd characteristic* (typically a large prime number) and in the second class the field is said to have *characteristic 2*. While at first sight this might be viewed as a somewhat technical distinction the choice of underlying field can have implications for both the security and the performance of the cryptosystem [Sim92]. This distinction is similar to one that is made between cryptosystems based on the discrete logarithm problem.

It is interesting to note that the problems of integer factorization and of discrete logarithms over a prime field appear to be of roughly the same difficulty. Techniques used to solve one problem can be adapted to tackle the other. As we have mentioned, there are elliptic curve analogs to RSA but it turns out that these are chiefly of academic interest since they offer essentially no practical advantages over RSA. This is primarily the case because elliptic curve variants of RSA actually rely for their security on the same underlying problem as RSA, namely that of integer factorization.

The situation is different with variants of discrete logarithm cryptosystems. The security of the elliptic curve variants of discrete logarithm cryptosystems depends on a restatement of the conventional discrete logarithm problem for elliptic curves. This restatement is such that current algorithms that solve the conventional discrete logarithm problem in what is termed *sub-exponential time* are of little value in attacking the analogous elliptic curve problem. Instead the only available algorithms for solving these elliptic curve problems are more general techniques that run in what is termed *exponential time*.

The distinction between exponential and sub-exponential time for solving some problem is a vitally important one. In essence it means that methods of finding a solution to one problem are becoming infeasible much faster than those for solving the other problem. As we will see, this has considerable practical significance.

In this note we will only be considering elliptic curve cryptosystems that depend for their security on the problem of taking elliptic curve discrete logarithms. In particular we will be considering analogs to the DSA and to the ElGamal encryption scheme. These will be described as the Elliptic Curve DSA (ECDSA) and the Elliptic Curve Encryption Scheme (ECES) respectively [IEEE97].

At a high level, we can already make one important statement. On a functional level, mechanisms for encryption or digital signatures can be devised so that they depend on any of the three types of problems we have already mentioned; integer factorization, conventional discrete logarithms, and elliptic curve discrete logarithms. There are however many trade-offs between the systems and these depend on many circumstances. It is the purpose of this note to give some guidance as to the implications of these potential differences.

## 3. Setting Up an Elliptic Curve Cryptosystem

In setting up any cryptosystem a certain amount of computation is required. In this section we will compare some of the basic set-up requirements for elliptic curve cryptosystems with those for users of RSA.

Recall that the elliptic curve cryptosystems of interest to us here are variants of the discrete logarithm cryptosystems like ElGamal and DSA. As a consequence certain parameters will be system parameters that are common to a set of users. Establishing the system parameters involves selecting an underlying finite field for the cryptosystem and a representation for the elements in the finite field. Then an "appropriate" elliptic curve has to chosen together with a point on the curve called the *generator*.

As in the conventional discrete logarithm case some finite fields, namely those of characteristic 2, appear to offer implementation and performance advantages in hardware. Unlike the discrete logarithm case[1], however, the choice of such a field does not appear to make the underlying problem any easier, at least as far as existing techniques are concerned, and so fields of this type will typically be the ones of choice.

There are several approaches for selecting an appropriate elliptic curve. They all tend to be mathematically very complicated and they have some limitations. It is perhaps worth pointing out at this stage that implementing elliptic curve cryptosystems can in fact be quite challenging without a good understanding of the mathematics of elliptic curves.

[1] When using a field of characteristic two for the discrete logarithm based cryptosystems there are opportunities for substantial performance improvements. However the discrete logarithm problem over such fields is somewhat easier to solve. Thus the potential benefits of using a field of characteristic 2 are consumed by the need for longer keys to attain the same level of security.

So we see that setting up the system parameters for an elliptic curve cryptosystem is quite involved. However, once it is done, the resulting elliptic curve parameters may be used for multiple users within a group (just as in the case of discrete logarithm cryptosystems) and each user has his or her public/private key pair. These key pairs are easy to generate and consist of a random, secret integer *k* that acts as the private key and that multiple of the generator point *G* on the curve that acts as the public key *kG* for the user. The security assumption is that it is hard to compute the private key *k* from the public key *kG*.

By way of comparison, the RSA cryptosystem requires no system parameters. The first stage of computing a public/private key pair consists of the user generating two primes of the appropriate size and computing the public modulus *n* as their product. This part of the computation can be rather computationally intensive (though not as intensive as setting up elliptic curve system parameters). The second stage for the user is then to compute the secret exponent *d*, or certain information that allows decryption to be optimized (so-called *Chinese Remainder Theorem* information), from what is usually a fixed public exponent *e*. The calculation of the secret exponent (or related information) is insignificant when compared to the time required to generate the primes. The various requirements for the different cryptosystems are given in Table 1 below.

| | **ECDSA and ECES** | **RSA** |
|---|---|---|
| **system parameters** | the field *F*, two field elements that represent the curve, the generator *G* on the curve and the order of *G* | none |
| **public key** | point $P= kG$ on the elliptic curve | modulus *n* and exponent *e* |
| **private key** | an integer *k* where $0 < k< q$ | exponent *d* or corresponding CRT information |

**Table 1:** System requirements for elliptic curve cryptosystems and RSA.

## 4. Practical Issues: Security

When we discuss the difficulty of solving hard problems, we normally do so in terms of the size of the problem facing the cryptanalyst. For RSA, the size of the problem is the length of the modulus that must be factored. For elliptic curve cryptosystems the size of the problem is the number of points *N* in the group we are working with. For the purposes of this note however, we will use an observation [Men95] that effectively equates this number of points directly with the size of the underlying field.

The elliptic curve discrete logarithm problem seems to be particularly hard to solve. Several algorithms might be used that have a running time that depends on the square root of *N* where *N* is the number of points in the group in which operations are performed.

It is interesting to note that such algorithms were among those used for factoring or solving the discrete logarithm problem when RSA was first proposed. The introduction of cryptosystems based on factoring and the discrete logarithm problem prompted developments in finding solutions to both problems. These improvements were the development of the *quadratic sieve*, described in [Sil87], and a further improvement with the *number field sieve* [BLP94]. The running time of these algorithms grows subexponentially in the size of the problem and for the size of RSA moduli that are typical today they are far superior for solving the problem than is the exponential Pollard Rho method [FR95].

In general the subexponential algorithms used to tackle the discrete logarithm problem cannot be adapted to the elliptic curve environment [Mil86]. There are, however, some exceptional cases where the elliptic curve discrete logarithm problem can be reduced to the conventional discrete logarithm problem (and hence becomes vulnerable to subexponential techniques) but these cases are readily classified and easily avoided [MOV91].

It appears that an elliptic curve cryptosystem implemented over the 160-bit field $GF(2^{160})$ currently offers roughly the same resistance to attack as would a 1024-bit RSA modulus[2] [Men95]. This currently offers the opportunity to use shorter keys than with RSA which might lead to better storage requirements and improved performance. We will address these issues in Section 6.

[2] A similar calculation suggests that an elliptic curve cryptosystem over a 136-bit field GF($2^{136}$) gives us roughly the same security as 768-bit RSA.

## 5. The state of academic interest and potential future development

| | | |
|---|---|---|
| 1977 | RSA proposed | |
| 1985 | Quadratic sieve factoring algorithm becomes increasingly practical | First use of elliptic curves proposed |
| 1991 | | Reduction of the elliptic curve discrete logarithm problem to a sub-exponential algorithm for some curves |
| 1993 | Number field sieve, an improved sub-exponential factoring algorithm | |
| 1994 | | Sub-exponential algorithm on high-genus hyperelliptic curves [ADH94] (no immediate implications to elliptic curves but an unexpected development) |

**Table 2**: Events in the evolution of RSA and elliptic curve cryptosystems.

In Table 2 we compare the evolution of factoring techniques with the development of elliptic curve cryptosystems.

Developments in the factoring problem have historically occurred more quickly than has progress in finding solutions to the elliptic curve discrete logarithm problem. There are however some important issues to consider.

From its publication RSA has been the public-key algorithm that has received most attention from implementors and analysts alike. For a great many years there were no competing proposals and so research has almost inevitably been focused on the problem of factoring. In addition, progress in factoring has been encouraged by such efforts as the compilation of the Cunningham Tables [BLS88] and the RSA Factoring Challenge sponsored by RSA Data Security, Inc. [FR95]. Another important issue that should not be overlooked is the apparent simplicity of the RSA algorithm. It is easy for most mathematicians to understand RSA and to understand the basic principles behind the major factoring techniques and so many researchers have contributed to an assessment of the security offered. The simplicity of RSA should be compared with what might appear to be the less approachable mathematics used in elliptic curve cryptosystems. This may well go some way to deterring some researchers from devoting their valuable time to what appears at first sight to be a very involved field.

The possibility that there will be advances in solving the elliptic curve discrete logarithm problem is a matter of speculation. Certainly the problem is currently harder than factorization for the same size problem but it is perhaps only now that the problem is coming under serious and wide-spread scrutiny. It will be particularly interesting to discover whether the choice of underlying field for the elliptic curve cryptosystem has any implications for security (as has become the case with the conventional discrete logarithm problem). Today, elliptic curve cryptosystems over a field of characteristic 2 are considered to offer implementational advantages but only time will tell whether the situation of the classical discrete logarithm problem is repeated in the case of elliptic curves, with some fields requiring larger system parameters for the same level of security.

## 6. Practical Issues: Implementation and Performance

Interest in elliptic curve cryptosystems is fueled by the appeal of basing a cryptosystem on a different hard problem and the fact that currently such a choice appears to lead to smaller system parameters and key sizes for the same level of security.

Throughout this section we will be comparing the requirements and performance of 1024-bit RSA (with public exponent $2^{16}+1$) with an elliptic curve cryptosystem implemented over the field *GF(q)* where *q* is 160 bits in length and the field is either of characteristic 2 or of odd characteristic. For the purposes of this note, we will assume that these different fields have essentially the same implementation requirements. In Table 3 we give a rough comparison of the storage requirements in bits for the schemes of interest to us in this note.

| | **ECDSA and ECES over GF(q)** | **RSA 1024-bit n and e=$2^{16}$+1** |
|---|---|---|
| **system parameters** | (4 x 160)+1 = 641 | 0 |
| **public key** | 160+1 = 161 | 1024 + 17 = 1041 |
| **private key** | 160 (801 with system parameters) | 2048 (or 2560 with CRT information) |

**Table 3:** The storage requirements in bits when making a naive comparison between an elliptic curve cryptosystem over *GF(q)*where *q* is 160 bits in length and RSA with a 1024-bit modulus.

With regard to the speed of implementation of these cryptosystems the situation is still very unclear. The basic elliptic curve operations are in fact quite complicated[3] (more complicated in fact than the operations required for RSA) and so if elliptic curve cryptosystems ever require the same size of parameters as does an implementation of RSA then the elliptic curve cryptosystem can be expected to be slower. In fact it is possible to envisage situations where even if the elliptic curve implementation uses smaller parameters than some implementation of RSA, the latter might remain the more efficient in terms of practical use. At present however, the current parameter advantages for elliptic curve cryptosystems are such that the speed of implementation can compare favorably with the performance of RSA.

Putting quantitative data into this part of the note is very difficult. We know of no figures or benchmarks with which we can compare an optimized version of RSA on one platform with an optimized version of some elliptic curve cryptosystem. However we make an attempt to qualitatively compare the performance of the various systems to the speed of RSA for the relevant operation and these results are presented in Table 4. These figures should be taken as a guide only and in making these comparisons we have assumed that one elliptic curve addition takes roughly the same effort as 10 modular multiplications. We feel that, for the purposes of this note, this figure will give a rough but fair comparison between cryptosystems. All techniques for precomputation that apply to discrete logarithm cryptosystems will apply equally to systems based on elliptic curves. It is interesting to note that even with the smaller keys required for elliptic curve cryptosystems, signature verification with RSA remains advantageous.

| | **ECDSA or ECES over GF(q)** | **RSA with 1024-bit n, e=216+1, and CRT** | **Discrete logarithm systems with 1024-bit prime** |
|---|---|---|---|
| **encryption** | 120 | 17 | 480 |
| **decryption** | 60 | 384 | 240 |
| **signing** | 60 | 384 | 240 |
| **verification** | 120 | 17 | 480 |

**Table 4:** The comparative performance of elliptic curve cryptosystems over *GF(q)* where *q* is 160 bits in length when compared with 1024-bit RSA and discrete logarithm cryptosystems for various cryptographic functions. Figures in the table are the

number of time units required to complete the given operation if we assume that one 1024-bit modular multiplication requires one unit of time. Not included in this table is that Diffie-Hellman key agreement requires 480 time units for each party. These figures do not take account of any of the various optimizations possible and they should be viewed as offering a rough comparison only.

[3] An elliptic curve operation involves a sequence of elliptic curve additions, and each addition consists of several arithmetic operations in the finite field. An RSA exponentiation involves a sequence of modular multiplications.

## 7. Practical Issues: Standards and Interoperability

Recently, elliptic curve cryptosystems have been considered as part of the efforts of various standards bodies including ANSI X9, ISO/IEC SC27 and the IEEE (P1363 Standard for Public-Key Cryptography). Elliptic curve cryptosystems are also included in OAKLEY, part of an Internet IETF draft for a key agreement protocol.

RSA is featured in many published and proposed standards worldwide, including all those previously mentioned with regards to elliptic curve cryptosystems and others besides such as ISO/IEC 9796, ANSI X9.31, Australian Standard 2805.5.3 to name but a few as well as the widely used, industry-sponsored, Public Key Cryptography Standards (PKCS). RSA is also used in many Internet initiatives and proposed protocols such as PEM, S/MIME, S/WAN, S-HTTP, and SSL. (See [Kal93] for a survey of encryption standards.) As far as commercial implementations are concerned RSA is used in a wide variety of software and hardware products and RSA licensees include Microsoft, AT&T, IBM, Motorola and many others too numerous to mention. See http://www.rsa.com/ for more details.

## 8. Conclusion

The opportunity to conveniently use elliptic curve cryptosystems within commercial applications is only now becoming a reality. There are however many issues to consider when making the choice between an application based on an elliptic curve cryptosystem and one based on RSA. In this note we have presented some of the issues (security, performance, standards and interoperability) that are perhaps most pertinent when making such a choice. The comparisons in this note are made, however, under the premise that an elliptic curve cryptosystem over *GF($2^{160}$)* offers the same security as 1024-bit RSA. Whether such an assumption will be realistic even some short distance into the future remains in interesting, but open, question.

**References**

[ADH94] L. M. Adleman, Jonathan DeMarrais and Ming-Deh Huang. A subexponential algorithm for discrete logarithms in the rational subgroup of the Jacobian of a hyperelliptic curve over a finite field. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*, pages 28-40, Springer-Verlag, 1994.

[BLS88] J. Brillhart, D. Lehmer, J. Selfridge, B. Tuckerman, and S. Wagstaff Jr. *Factorizations of bn 1, b* = 2, 3, 5, 6, 7, 10, 11, 12 *up to high powers*. Volume 22 of *Contemporary Mathematics,* American Mathematical Society, 1988.

[BLP94] P. Buhler, H.W. Lenstra, and C. Pomerance. The development of the number field sieve. Volume 1554 of *Lecture Notes in Computer Science,* Springer-Verlag, 1994.

[DH76] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory,* IT-22: 644-654, 1976.

[Dem94] N. Demytko. A new elliptic curve based analogue of RSA. In *Advances in Cryptology - Eurocrypt'93,* pages 40-49, Springer-Verlag, 1994.

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31: 469-472, 1985.

[FR95] P. Fahn and M.J.B. Robshaw. *Results from the RSA Factoring Challenge.* Technical Report TR-501, version 1.3, RSA Laboratories, January 1995.

[IEEE97] IEEE Working Group P1363. Working Draft: Standard for RSA, Diffie-Hellman and related public-key cryptography. Editorial contribution. March 1997.

[Kal93] B.S. Kaliski Jr. A survey of encryption standards. *IEEE Micro*, 13(6): 74-81, December 1993.

[Kob87] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation,* 48:203-209, 1987.

[KMOV92] K. Koyama, U.M. Maurer, T. Okamoto, and S.A. Vanstone. New public-key schemes based on elliptic curves over the ring *Zn*. In *Advances in Cryptology - Crypto'91,* pages 40-49, Springer-Verlag, 1992.

[Len87] H.W. Lenstra, Jr. Factoring integers with elliptic curves. *Annuals of Mathematics*, 126: 649-673, 1987.

[Men93] A. Menezes. *Elliptic Curve Public Key Cryptosystems.* Kluwer Academic Publishers, 1993.

[Men95] A. Menezes. Elliptic Curve Cryptosystems. *CryptoBytes,* Vol.1 No.2, Summer 1995.

[MOV91] A. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 80-89, ACM, 1991.

[Mil86] V.S. Miller. Use of elliptic curve in cryptography. In *Advances in Cryptology - Crypto'85*, pages 417-426, Springer-Verlag, 1986.

[NIST94] National Institute of Standards and Technology (NIST). *FIPS Publication 186: Digital Signature Standard,* May 19, 1994.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2): 120-126, February 1978.

[Sil87] R.D. Silverman. The multiple polynomial quadratic sieve. *Mathematics of Computation,* 48: 329-339, 1987.

[Sim92] G.J. Simmons, editor. *Contemporary Cryptology - The Science of Information Integrity*. IEEE Press, 1992.

RSA Home | ECC Central
Mathematician s Intro to ECC| Recommendations on ECC
The Experts Comment on ECC | ECC & RSA s BSAFE 4.0
Q&A on ECC | BSAFE 4.0 Home Page

Website questions or comments: webmaster@rsa.com