The Elliptic Curve Cryptosystem

REMARKS ON THE SECURITY OF
THE ELLIPTIC CURVE CRYPTOSYSTEM

Published: September 1997

Updated: July 2000

A Certicom Whitepaper

The Elliptic Curve Cryptosystem (*ECC*) *provides the highest strength-per-bit of any cryptosystem known today. This paper provides an overview of the three hard mathematical problems which provide the basis for the security of public-key cryptosystems used today*: *the integer factorization problem* (*IFP*), *the discrete logarithm problem* (*DLP*), *and the elliptic curve discrete logarithm problem* (*ECDLP*). *An overview of current hardware and software attacks on ECDLP is also provided.*

Contents

Please send comments and inquiries to:

Certicom Corp.
5520 Explorer Drive 4th Floor
Mississauga, Ontario, Canada L4W 5L1
Phone: (905) 507-422
FAX: (905) 507-4230
info@certicom.com
http://www.certicom.com/

1. Introduction

Since the invention of public-key cryptography in 1976 by Whitfield Diffie and Martin Hellman [6, numerous public-key cryptographic systems have been proposed. All of these systems based their security on the difficulty of solving a mathematical problem.

Over the years, many of the proposed public-key cryptographic systems have been broken and many others have been demonstrated to be impractical. Today, only three types of systems are considered both secure and efficient. Examples of such systems and the mathematical problems on which their security is based, are:

1. **Integer factorization problem (IFP):** RSA and Rabin-Williams.
2. **Discrete logarithm problem (DLP):** the U.S. government's Digital Signature Algorithm (DSA), the Diffie-Hellman key agreement scheme, the ElGamal encryption and signature schemes, the Schnorr signature scheme, and the Nyberg-Rueppel signature scheme.
3. **Elliptic curve discrete logarithm problem (ECDLP):** the elliptic curve analog of the DSA (ECDSA), and the elliptic curve analogs of the Diffie-Hellman key agreement scheme, the ElGamal encryption and signature schemes, the Schnorr signature scheme, and the Nyberg-Rueppel signature scheme.

It must be emphasized that none of these problems have been *proven* to be intractable (i.e., difficult to solve in an efficient manner). Rather, they are *believed* to be intractable because years of intensive study by leading mathematicians and computer scientists has failed to yield efficient algorithms for solving them.

The objective of this paper is to foster a greater understanding among information security experts of the security of the **Elliptic Curve Cryptosystem (ECC)**. An overview of the state-of-the-art algorithms for solving the three mathematical problems is given. An overview of current hardware and software attacks on ECDLP, on which ECC is based, is also provided. A more in-depth analysis of the ECDLP aimed towards the advanced reader will be provided in a forthcoming Certicom white paper.

2. The Integer Factorization Problem (IFP)

2.1 Problem definition

The **integer factorization problem (IFP)** is the following: given a composite number n that is the product of two large prime numbers p and q, find p and q.

While finding large prime numbers is a relatively easy task, the problem of factoring the product of two such numbers is considered computationally intractable if the primes are carefully selected. Based on the difficulty of this problem, Rivest, Shamir and Adleman [27] developed the RSA public-key cryptosystem. Another public-key cryptosystem whose security lies on the intractability of IFP is due to Rabin and Williams [26, 34].

While the integer factorization problem has received some attention over the centuries from well-known mathematicians like Fermat and Gauss, it is only in the past 20 years that significant progress has been made towards its resolution. There are two main reasons for this phenomenon. First, the invention of the RSA cryptosystem in 1978 stimulated many mathematicians to study the problem. And second, high-speed computers became available for the implementation and testing of sophisticated algorithms. Fermat and Gauss would have had little incentive for inventing the number field sieve factoring algorithm (see Section 2.2) since this algorithm is more cumbersome than trial division for the purpose of factoring integers by hand.

2.2 Known attacks

There are basically two types of factoring algorithms, **special-purpose** and **general-purpose**. **Special-purpose** factoring algorithms attempt to exploit special features of the number n being factored. In contrast, the running times of **general-purpose** factoring algorithms depend only on the size of n.

One of the most powerful special-purpose factoring algorithms is the elliptic curve factoring method (ECM) that was invented in 1985 by Hendrik Lenstra Jr. [16]. The running time of this method depends on the size of the prime factors of n, and hence the algorithm tends to find small factors first. The ECMNET project, started in January 1998 to find large factors by ECM, has been successful in locating factors of 50 digits (166 bits) or more. The largest prime factor found thus far by ECM is a 54-digit (180-bit) factor of a 127-digit (422-bit) number; the computation was carried out by N. Lygeros and M. Mizony and reported on December 26, 1999.

Just prior to the development of the RSA cryptosystem, the best general-purpose factoring algorithm was the **continued fraction algorithm** [19], which could factor numbers up to 40 decimal digits (133 bits). This algorithm was based on the idea of using a **factor base** of primes and generating an associated set of linear equations whose solution ultimately led to a factorization. This is the same idea underlying the best general-purpose algorithms used today: the **quadratic sieve (QS)** and the **number field sieve (NFS).** Both these algorithms can be easily parallelized to permit factoring on distributed networks of workstations. Large mainframe computers or supercomputers are therefore not essential to factor large numbers.

The quadratic sieve was developed by Carl Pomerance in 1984 [25]. Initially, it was used to factor numbers in the 70-decimal digit (233-bit) range. In 1994, it was used by a group of researchers led by Arjen Lenstra [1] to factor the 129-decimal digit (429-bit) RSA challenge number that was posed by Martin Gardner in 1977 [10]. The factorization was carried out in 8 months by about 1600 computers around the world. The total running time for the factorization was estimated to be 5000 MIPS years.

The number field sieve as first developed in 1989 [15] works best on numbers of a special form. The algorithm was used to factor the 155-decimal digit (513-bit) number $2^{512} + 1$. It was subsequently extended (see [3]) to a general-purpose factorization algorithm. While initially thought to be slower in practice than the quadratic sieve for factoring integers having fewer than 150 decimal digits (500 bits), recent experiments have suggested that the NFS is indeed the superior algorithm for factoring integers having at least 120 decimal digits (400 bits). In 1996, a group led by Arjen Lenstra [4] used the NFS to factor a 130-decimal digit (432-bit) number. The factorization was estimated to take less than 15% of the 5000 MIPS years that was required for the factorization of the 129-decimal digit RSA challenge number. The authors concluded that

factoring a 512-bit (155 decimal-digit) number could take less than 5 times this effort. On August 22, 1999, a group of international researchers announced the factorization of the 155-digit RSA Challenge Number; using the generalized number field sieve, the total running time was approximately 8000 MIPS years, with a calendar time of 3.7 months. Table 1 contains some historical data on the progress of integer factorization.

Year	Number of decimal digits	Number of bits	MIPS years
1984	71	236	0.1
1988	106	352	140
1993	120	399	825
1994	129	429	5000
1995	119	395	250
1996	130	432	750
1999	140	466	2000
1999	155	512	8000

Table 1: Historical data on the integer factorization problem.

These results indicate that a 512-bit modulus n provides only marginal security when used in the RSA cryptosystem. For long-term security, 1024-bit or larger moduli should be used.

3. The Discrete Logarithm Problem (DLP)

3.1 Problem definition

If p is a prime number, then Z_p denotes the set of integers $\{0, 1, 2, \ldots, p - 1\}$, where addition and multiplication are performed modulo p. It is well-known that there exists a non-zero element $\alpha \in Z_p$ such that each non-zero element in Z_p can be written as a power of α; such an element α is called a **generator** of Z_p.

The **discrete logarithm problem (DLP)** is the following: given a prime p, a generator α of Z_p, and a non-zero element $\beta \in Z_p$, find the unique integer l, $0 \leq l \leq p - 2$, such that $\beta \equiv \alpha^l \pmod p$. The integer l is called the **discrete logarithm** of β to the base α.

Based on the difficulty of this problem, Diffie and Hellman [6] proposed the well-known Diffie-Hellman key agreement scheme in 1976. Since then, numerous other cryptographic protocols whose security depends on the DLP have been proposed, including: the U.S. government digital signature algorithm (DSA) [8], the ElGamal encryption and signature schemes [7], the Schnorr signature scheme [32], and the Nyberg-Rueppel signature scheme [20]. Due to interest in these applications, the DLP has been extensively studied by mathematicians for the past 20 years.

3.2 Known attacks

As with the integer factorization problem, there are two types of algorithms for solving the discrete logarithm problem. **Special-purpose** algorithms attempt to exploit special features of

the prime p. In contrast, the running times of **general-purpose** algorithms depend only on the size of p.

The fastest general-purpose algorithms known for solving the DLP are based on a method called the **index-calculus.** In this method, a database of small primes and their corresponding logarithms is constructed, subsequent to which logarithms of arbitrary field elements can be easily obtained. This is reminiscent of the factor base methods for integer factorization. For this reason, if an improvement in the algorithms for either the IFP or DLP is found, then shortly after a similar improved algorithm can be expected to be found for the other problem. As with the factoring methods, the index-calculus algorithms can be easily parallelized.

As in the case with factoring, the best current algorithm known for the DLP is the **number field sieve** [11, 26]. It has precisely the same asymptotic running time as the corresponding algorithm for integer factorization. This can loosely be interpreted as saying that finding logarithms in the case of a k-bit prime modulus p is roughly as difficult as factoring a k-bit composite number n.

The implementation of discrete logarithm algorithms has lagged behind the analogous efforts for factoring integers. In 1990, Brian LaMacchia and Andrew Odlyzko [14] used a variant of the index-calculus method called the **Gaussian integer method** to compute discrete logarithms modulo a 191-bit prime.

In 1998, Weber and Denny announced the solution to the McCurley Diffie-Hellman Challenge, which involved solving a discrete logarithm problem modulo a 129-digit prime. The prime number used was of special form, hence the number field sieve was employed, and was particularly effective for this challenge. The results were presented at Crypto '98 [5].

These results indicates for long-term security, 1024-bit or larger moduli p should be used in discrete logarithm cryptosystems.

4. The Elliptic Curve Discrete Logarithm Problem (ECDLP)

4.1 Problem definition

If q is a prime power, then F_q denotes the finite field containing q elements. In applications, q is typically a power of 2 (2^m) or an odd prime number (p).
The **elliptic curve discrete logarithm problem (ECDLP)** is the following: given an elliptic curve E defined over F_q, a point $P \in E(\mathsf{F}_q)$ of order n, and a point $Q \in E(\mathsf{F}_q)$, determine the integer l, $0 \le l \le n - 1$, such that $Q = lP$, provided that such an integer exists.

Based on the difficulty of this problem, Neal Koblitz [12] and Victor Miller [18] independently in 1985 proposed using the group of points on an elliptic curve defined over a finite field to implement the various discrete log cryptosystems. One such cryptographic protocol that is being standardized by accredited standards organizations is the elliptic curve analog of the DSA, called ECDSA.

The following comments concern some commonly held misconceptions about the history of IFP versus ECDLP. It is often stated that IFP is easier to state and understand than the ECDLP.

While this statement is true, it erroneously leads people to believe that the IFP has been *seriously* studied by a multitude of people over thousands of years. This is indeed false. As indicated in Section 2.2, there have been only two major advances in techniques for solving the IFP: the quadratic sieve factoring algorithm (together with its predecessor, the continued fraction factoring algorithm), and the number field sieve. The latter algorithm involves some sophisticated mathematics (especially algebraic number theory), and is only completely understood by a small community of number theorists. And, as mentioned in Section 2.1, prior to the advent of the computer age, mathematicians were constrained to looking for algorithms for IFP that were efficient by *hand* rather than on large networks of computers. Another fact that is commonly overlooked is that much of the work done on the DLP prior to 1985 also applies to the ECDLP – that is because the ECDLP can be viewed as being the same as the DLP but in a different algebraic setting.

4.2 Known attacks

Since 1985, the ECDLP has received considerable attention from leading mathematicians around the world. An algorithm due to Pohlig and Hellman [23] reduces the determination of l to the determination of l modulo each of the prime factors of n. Hence, in order to achieve the maximum possible security level, n should be prime. The best algorithm known to date for the ECDLP in general is the Pollard rho-method [24] which, with the speed up proposed by Gallant, Lambert and Vanstone [9], and Wiener and Zuccherato [35], takes about $\sqrt{\pi n}/2$ steps, where a *step* here is an elliptic curve addition. In 1993, Paul van Oorschot and Michael Wiener [22] showed how the Pollard rho-method can be parallelized so that if r processors are used, then the expected number of steps by each processor before a single discrete logarithm is obtained is $\sqrt{\pi n}/2r$. Most significantly, no index-calculus-type algorithms are known for the ECDLP as for the DLP. For this reason, the ECDLP is believed to be much harder than either the IFP or the DLP in that no subexponential-time general-purpose algorithm is known.

In 1991, Menezes, Okamoto and Vanstone (MOV) [17] showed how the ECDLP can be reduced to the DLP in extension fields of F_q, where the index-calculus methods can be applied. However, this *MOV reduction algorithm* is only efficient for a very special class of curves known as *supersingular curves.* Moreover, there is a simple test to ensure that an elliptic curve is not vulnerable to this attack. Supersingular curves are specifically prohibited in all (draft) standards of elliptic curve systems such as IEEE P1363, ANSI X9.62, and ANSI X9.63.

Another weak class of elliptic curves are the so-called *anomalous curves* – these are curves E defined over F_q which have exactly q points. The attack on these curves was discovered independently by Semaev [30], Smart [31], and Satoh and Araki [29], and generalized by R□k [28]. As with supersingular curves, there is a simple test to ensure that an elliptic curve is not vulnerable to this atack; through this test, these curves are specifically prohibited in all (draft) standards of elliptic curve systems.

Koblitz curves [13] are a class of elliptic curves over F_{2^m} whose defining equations have coefficients in F_2. Koblitz curves are shown to be very efficient in computing kP for arbitrary k and any point P on the curve, and hence very attractive for cryptographic use. Gallant, Lambert and Vanstone [9], and Wiener and Zuccherato [35], showed that computing elliptic curve logarithms for such curves in $E(\mathrm{F}_{2^m})$ can be sped up by a factor of $\sqrt{m}$. This speed up is

reflected in the recent solution to the Certicom ECC Challenge ECC2K-108. However, when *m* is large enough (*m*>160), the speed up is still insignificant compared to the amount of computation required with present day technology.

There has been ongoing research on the ECDLP and related problems. The Third Workshop on Elliptic Curve Cryptography was held at the University of Waterloo in 1999, and a fourth workshop is to be staged at the University of Essen, Germany, in October, 2000. Numerous papers were also published in various journals and presented in cryptography-related conferences.

Software Attacks

According to the Certicom ECC Challenge document, we estimate that a Pentium 100 can perform 16,000 operations per second for a curve over $F_{2^{89}}$. This would then require 15,550 days for a single computer running 24 hours a day to find a single discrete logarithm. A network of 3,000 such machines would require about 5 days. This figure translates to about 80 iterations per second on a 1 MIPS machine.

Table 2 shows the computing power required to compute a single discrete logarithm using the Pollard rho-method for various values of *n.*

Field size (in bits)	Size of *n* (in bits)	$\sqrt{\pi n}/2$	MIPS years
163	160	2^{80}	9.6×10^{11}
191	186	2^{93}	7.9×10^{15}
239	234	2^{117}	1.6×10^{23}
359	354	2^{177}	1.5×10^{41}

Table 2: Computing power required to compute elliptic curve logarithms with the Pollard rho-method.

As an example, if 10,000 computers each rated at 1,000 MIPS are available, and $n \approx 2^{160}$, then an elliptic curve discrete logarithm can be computed in 96,000 years. Andrew Odlyzko [21] has estimated that if 0.1% of the world's computer power were available for one year to work on a collaborative effort to break some challenge cipher, then the computing power available would be 10^8 MIPS years in 2004 and 10^{10} to 10^{11} MIPS years in 2014.

To put the numbers in Table 2 into perspective, Table 3 (due to Odlyzko [21]) shows the estimated computing power required to factor integers with current versions of the number field sieve. This is also roughly equal to the time it takes to compute discrete logarithms modulo a prime *p* of the same bitlength as *n*.

Size of integer to be factored (in bits)	MIPS years
512	3×10^4
768	2×10^8

1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

Table 3: Computing power required to factor integers using the general number field sieve.

Hardware Attacks

A more promising attack (for well-funded attackers) on elliptic curve systems would be to build special-purpose hardware for a parallel search using the Pollard rho-method. Van Oorschot and Wiener [22] provide a detailed study of such a possibility. In their 1994 study, they estimated that if $n \approx 10^{36} \approx 2^{120}$, then a machine with m = 325,000 processors that could be built for about US$10 million would compute a single discrete logarithm in about 35 days. However, if $n > 2^{160}$, then these attacks are infeasible.

Discussion

It should be pointed out that in the software and hardware attacks previously described, computation of a single elliptic curve discrete logarithm has the effect of revealing a *single* user's private key. The same effort must be repeated in order to determine another user's private key.

Blaze et al. [2] reported on the minimum key lengths required for secure symmetric-key encryption schemes (such as DES and IDEA). Their report provides the following conclusion:

> *To provide adequate protection against the most serious threats – well-funded commercial enterprises or government intelligence agencies – keys used to protect data today should be at least 75 bits long. To protect information adequately for the next 20 years in the face of expected advances in computing power, keys in newly deployed systems should be at least 90 bits long.*

Extrapolating these conclusions to the case of elliptic curves, we see that n should be at least 150 bits for short-term security and at least 180 bits for medium-term security. This extrapolation is justified by the following considerations:

1. Exhaustive search through a k-bit symmetric-key cipher takes about the same time as the Pollard rho-algorithm applied to an elliptic curve having a $2k$-bit parameter n.
2. Exhaustive searches with a symmetric-key cipher and the Pollard rho-algorithm can be parallelized with a linear speedup.
3. A basic operation with elliptic curves (addition of two points) is computationally more expensive than a basic operation in a symmetric-key cipher (encryption of one block).
4. In both symmetric-key ciphers and elliptic curve systems, a "break" has the same effect: it recovers a single private key.

4.3 Choice of underlying field $\mathbf{F}_q$ and elliptic curve E.

When setting up an elliptic curve cryptosystem, there are three basic decisions that need to be made:

1. Selection of the underlying finite field $\mathbf{F}_q$.
2. Selection of the representation for the elements of $\mathbf{F}_q$.
3. Selection of the elliptic curve E over $\mathbf{F}_q$.

The following remarks discuss how the choices of the underlying field, its representation, and elliptic curve, may impact upon the intractability of the ECDLP (and hence on the security of the elliptic curve cryptosystem).

1. The two most common choices in practical applications for the underlying finite field are $\mathbf{F}_{2^m}$ and $\mathbf{F}_p$ (where p is an odd prime). The ECDLP is equally difficult for instances which use $\mathbf{F}_{2^m}$ as those which use $\mathbf{F}_p$, and where the sizes 2^m and p of the fields are approximately equal. There have not been any mathematical discoveries to date which suggest that the ECDLP for elliptic curves over $\mathbf{F}_{2^m}$ may be any easier or harder than the ECDLP for elliptic curves over $\mathbf{F}_p$.
2. If the field $\mathbf{F}_{2^m}$ is selected as the underlying finite field, then there are many ways in which the elements of $\mathbf{F}_{2^m}$ can be represented. The two most efficient ways are an *optimal normal basis representation* and a *polynomial basis representation*. Since elements in one representation can be efficiently converted to elements in the other representation by using an appropriate change-of-basis matrix, the intractability of the ECDLP is not affected by the choice of representation.

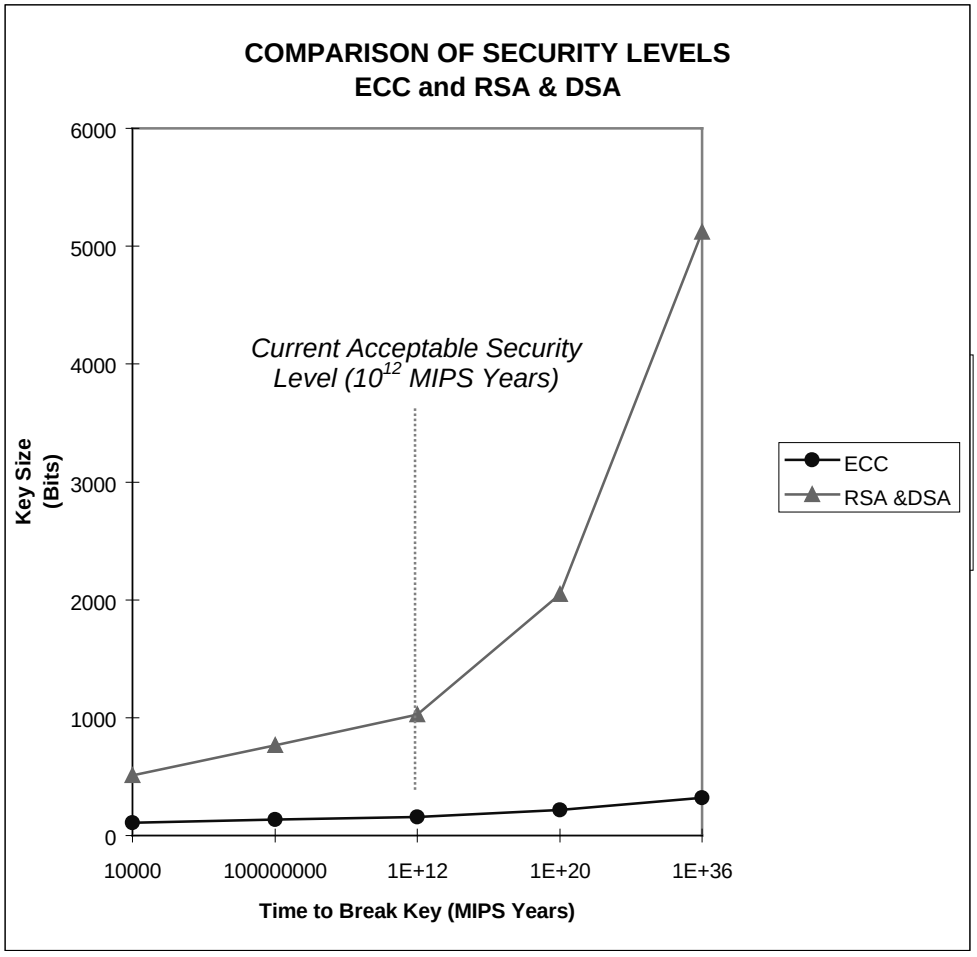

Figure 1: Comparison of Security Levels

3. As mentioned in Section 4.2, the MOV reduction algorithm yields an algorithm for the ECDLP when the elliptic curve is supersingular. Most elliptic curves, however, are non-supersingular. Moreover, it can easily be verified whether or not the MOV reduction algorithm is feasible for a given elliptic curve – hence this attack is easily avoided in practice. Also, since it can easily be detected whether a given curve is anomalous, the attack on anomalous curve is easily avoided. When selecting a (non-anomalous) non-supersingular elliptic curve, one may select a curve at random, or one may select a curve with special properties which may result in faster elliptic curve arithmetic. An example of a special class of curves that has been proposed are the *Koblitz curves* (see Koblitz [13] and Solinas [33]). Though it has been shown that there is a linear speedup on attacking Koblitz curves on multiple machines, this speedup is not significant with present-day technology for a large enough field.

5. Comparison

Given the current state of our knowledge about algorithms for the IFP, DLP and ECDLP problems, we can conclude that the ECDLP is significantly more difficult than either the IFP or

the DLP. Figure 1 compares the time required to solve an instance of the ECDLP (and hence break ECC) with the time required to solve instances of the IFP or DLP (and hence break RSA or DSA, respectively) for various modulus sizes and using the best general algorithms known. The running times are computed in MIPS years. As a benchmark, it is generally accepted that 10^{12} MIPS years represents reasonable security at this time. In Figure 1, the times to break RSA and DSA are grouped together because the best algorithms known for IFP and DLP have approximately the same asymptotic running times.

From Figure 1, we see that to achieve reasonable security, RSA and DSA should employ 1024-bit moduli, while a 160-bit modulus should be sufficient for ECC. Moreover, the security gap between the systems increases dramatically as the moduli sizes increases. For example, 300-bit ECC is dramatically more secure than 2048-bit RSA or DSA.

6. Conclusions

A comparison of the three hard mathematical problems on which the well-known public-key cryptosystems are based clearly highlights the fact that none of these are provably intractable. Years of intensive study has resulted in a widely held view that ECDLP is significantly more difficult than either the IFP or the DLP. The general conclusion of leading cryptographers is that the ECDLP in fact requires fully exponential time to solve. Based on this research and their own cryptographic expertise, industry leaders have accepted the Elliptic Curve Cryptosystem as a mature technology and are now implementing it for widespread deployment.

7. References

[1] D. Atkins, M. Graff, A.K. Lnstra and P.C. Leyland, "The magic words are SQUEAMISH OSSIFRAGE", *Advances in Cryptology – ASIACRYPT '94*, Lecture Notes in Computer Science, volume 917, Springer-Verlag, pages 263-277, 1995.

[2] Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Wiener, "Minimal key lengths for symmetric ciphers to provide adequate commercial security", January 1996, available from `http://theory.lcs.mit.edu/~rivest/publications.html`

[3] J. P. Buhler, H.W. Lenstra Jr. and C. Pomerance, "Factoring integers with the number field sieve", in *The Development of the Number Field Sieve*, Lecture Notes in Computer Science, volume 1554, Springer-Verlag, pages 11-42, 1993.

[4] J. Crowie, B. Dodson, R. Elkenbracht-Huizing, A. Lenstra, P. Montgomery and J. Zayer, "A world wide number field sieve factoring record: on to 512 bits", *Advances in Cryptology - ASIACRYPT '96*, Lecture Notes in Computer Science, volume 1163, Springer-Verlag, pages 382-394, 1996.

[5] T. Denny, D. Weber, "The Solution of McCurley's Discrete Log Challenge", *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science, colume 1462, Springer-Verlag, pages 458-471, 1998.

[6] W. Diffie and M. Hellman, "New Directions in cryptography", *IEEE Transactions on Information Theory,* volume 22, pages 644-654, 1976.

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

[8] FIPS 186, "Digital signature standard", National Institute for Standards and Technology, 1993. Available from `http://csrc.ncsl.nist.gov/fips/`

[9] R. Gallant, R. Lambert and S. Vanstone, "Improving the parallelized Pollard lambda search on binary anomalous curves", to appear in *Mathematics of Computation*.

[10] M. Gardner, "A new kind of cipher that would take millions of years to break", *Scientific American*, volume 237, pages 120-124, August 1977.

[11] D. Gordon, "Discrete logarithms in *GF(p)* using the number field sieve", *SIAM Journal on Discrete Mathematics,* volume 6, pages 124-138, 1993.

[12] N. Koblitz, "Elliptic curve cryptosystems", *Mathematics of Computation*, Volume 48, pages 203-209, 1987.

[13] N. Koblitz, "CM-curves with good cryptographic properties", *Advances in Cryptology – CRYPTO '91*, Lecture Notes in Computer Science, volume 576, Springler-Verlag, pages 279-287, 1992.

[14] B.A. LaMacchia and A.M. Odlyzko, "Computation of discrete logarithms in prime fields", *Designs, Codes and Cryptography*, volume 1, pages 47-62, 1991.

[15] A.K. Lenstra, H.W. Lenstra Jr., M.S. Manasse and J.M. Pollard, "The number field sieve", in *The Development of the Number Field Sieve*, Lecture Notes in Mathematics, volume 1554, Springer-Verlag, pages 11-42, 1993.

[16] H.W. Lenstra, "Factoring integers with elliptic curves", *Annals of Mathematics*, volume 126, pages 649-673, 1987.

[17] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Transactions on Information Theory*, volume 39, pages 1639-1646, 1993.

[18] V. Miller, "Uses of elliptic curves in cryptography", *Advances in Cryptology CRYPTO '85*, Lecture Notes in Computer Science, volume 218, Springer-Verlag, pages 417-426, 1986.

[19] M.A. Morrison and J. Brillhart, "A method of factoring and the factorization of F_7", *Mathematics of Computation*, volume 29, pages 183-205, 1975.

[20] K. Nyberg and R. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem", *Designs, Codes and Cryptography,* volume 7, pages 61-81, 1996.`

[21] A. Odlyzko, "The future of integer factorization", CryptoBytes - The technical newsletter of RSA Laboratories, volume 1, number 2, pages 5-12, Summer 1995. Also available from `http://www.rsa.com/`

[22] P. van Oorschot and M. Wiener, "parallel collision search with cryptanalytic applications", to appear in *Journal of Cryptology*. An earlier version appeared in the Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, November 2-4, 1994, pages 210-218.

[23] S. Pohlig and M. Hellman, "An improved algorithm for computing logarithms over *GF(p)* and its cryptographic significance", *IEEE Transaction on Information Theory*, volume 24, pages 106-110, 1978.

[24] J. Pollard, "Monte Carlo methods for index computation mod *p*", *Mathematics of Computation*, volume 32, pages 918-924, 1978.

[25] C. Pomerance, "The quadratic sieve factoring algorithm", *Advances in Cryptology EUROCRYPT '84*, Lecture Notes in Computer Science, volume 209, Springer-Verlag, pages 169-182, 1985.

[26] M.O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization", MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[27] R.L. Rivest, A. Shamir and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, volume 21, pages 120-126, 1978.

[28] H. R□k, `On the discrete logarithm in the divisor class group of curves`, preprint, 1997.

[29] T. Satoh and K. Araki, "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves", *Commentarii Math. Univ. St. Pauli*, volume 47, pages 81-92, 1998.

[30] I. Semaev, "Evaluation of discrete logarithm on some elliptic curves", to appear in *Mathematics of Computation*.

[31] N. Smart, Announcement of an attack on the ECDLP for anomalous elliptic curves, 1997.

[32] C.P. Schnorr, "Efficient signature generation by smart cards", *Journal of Cryptology*, volume 4, pages 161-174, 1991.

[33] J. Solinas, "An improved algorithm for arithmetic on a family of elliptic curves", *Advances in Cryptology – CRYPTO '97*, Lecture Notes in Computer Science, volume 1294, Springer-Verlag, pages 357-371, 1997.

[34] H.C. Williams, "A modification of the RSA public-key encryption procedure", *IEEE Transactions on Information Theory*, volume 26, pages 726-729, 1980.

[35] M. Wiener and R. Zuccherato, "Faster attacks on elliptic curve cryptosystems", *Selected Areas in Cryptography*, Lecture Notes on Computer Science, 1556 (1999), Springer-Verlag, 190-200.