A Digital Signature Scheme Based on Two Hard Problems

Dimitrios Poulakis and Robert Rolland

Abstract In this paper we propose a signature scheme based on two intractable problems, namely the integer factorization problem and the discrete logarithm problem for elliptic curves. It is suitable for applications requiring long-term security and provides smaller signatures than the existing schemes based on the integer factorization and integer discrete logarithm problems.

Keywords: Digital signature • Integer factorization • Elliptic curve discrete logarithm • Supersingular elliptic curves • Pairing • Map to point function • Long-term security

1 Introduction

Many applications of the Information Technology, such as encryption of sensitive medical data or digital signatures for contracts, need long-term cryptographic security. Unfortunately, today's cryptography provides strong tools only for short-term security [5]. Especially, digital signatures do not guarantee the desired long-term security. In order to achieve this goal Maseberg [20] suggested the use of more than one sufficiently independent signature schemes. Thus, if one of them is broken, then it can be replaced by a new secure one. Afterward the document has to be re-signed. Again we have more than one valid signatures of our document. Of course, a drawback of the method is that the document has to be re-signed.

In order to avoid this problem, it may be interesting for applications with long-term, to base the security of cryptographic primitives on two difficult problems, so if any of these problems is broken, the other will still be valid and hence the signature will be protected. We propose in this paper an efficient signature scheme built taking

D. Poulakis (✉)
Department of Mathematics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
e-mail: poulakis@math.auth.gr

R. Rolland
Institut de Mathématiques de Marseille, Université d'Aix-Marseille, Case 907,
13288 Marseille Cedex 9, France
e-mail: robert.rolland@acrypta.fr

N.J. Daras, M.Th. Rassias (eds.), *Computation, Cryptography, and Network Security*, DOI 10.1007/978-3-319-18275-9_19

into account this constraint. The following signature scheme is based on the integer factorization problem and the discrete logarithm problem on a supersingular elliptic curve. Remark that these two problems have similar resistance to attack, thus they can coexist within the same protocol. The use of a supersingular curve allows us to easily build a pairing that we use to verify the signature.

Several signature schemes combining the intractability of the integer factorization problem and integer discrete logarithm problem were proposed but they have proved either to be enough to solve the one of two problems for breaking the system or to have other security problems [6, 9, 16–19, 22, 27]. An interesting scheme based on the above problems is GPS [8]. Furthermore, some recent such schemes are given in [12, 13, 19, 24, 25, 27].

In Sect. 2 we describe the infrastructure for the implementation of the scheme. Then we present the key generation, the generation of a signature and the verification. In Sect. 3 we show how to build an elliptic curve adapted to the situation and how to define a valuable pairing on it. In Sect. 4 we address the problem of the map to point function and give a practical solution. We deal with the performance of our scheme and compare it with others in Sect. 5. In Sect. 6 we give a complete example that shows that the establishment of such a system can be made in practice. In Sect. 7 we study the security of the scheme. Finally Sect. 8 concludes the paper.

2 The Proposed Signature Scheme

In this section we present our signature scheme.

2.1 Public and Private Key Generation

A user $\mathscr{A}$, who wants to create a public and a private key selects:

1. primes p_1 and p_2 such that the factorization of $n = p_1 p_2$ is unfeasible;
2. an elliptic curve E over a finite field $\mathbb{F}_q$, a point $P \in E(\mathbb{F}_q)$ with $\mathrm{ord}(P) = n$ and an efficiently computable pairing e_n such that $e_n(P, P)$ is a primitive nth root of 1;
3. $g \in \{1, \dots, n-1\}$ with $\gcd(g, n) = 1$, $a \in \{1, \dots, \phi(n) - 1\}$ and computes $Q = g^a P$;
4. two one-way, collision-free hash functions, $h : \{0,1\}^* \rightarrow \{0, \dots, n-1\}$ and $H : \{0,1\}^* \rightarrow < P >$, where $< P >$ is the subgroup of $E(\mathbb{F}_q)$ generated by P.

$\mathscr{A}$ publishes the elliptic curve E, the pairing e_n, and the hash functions h and H. The public key of $\mathscr{A}$ is (P, Q, g, n) and his private key (a, p_1, p_2).

2.2 Signature Generation

The user $\mathscr{A}$ wants to sign a message $m \in \{0,1\}^*$. Then he chooses at random $k, l \in \{1, \dots, \phi(n) - 1\}$ such that $k + l = a$. Next, he computes

$$s = k + h(m) + n \bmod \phi(n) \quad \text{and} \quad S = g^l H(m).$$

Let $x(S)$ be the x-coordinate of S and b a bit determining S. The signature of m is $(s, x(S), b)$.

2.3 Verification

Suppose that (s, x, b) is the signature of m. The receiver uses b in order to determine y such that $S = (x, y)$ is a point of $E(\mathbb{F}_q)$. He accepts the signature if and only if

$$e_n(g^s P, S) = e_n(g^{h(m)+n} Q, H(m)).$$

Proof of Correctness of Verification. Suppose that the signature (x, s, b) is valid and $S = (x, y)$ is a point of $E(\mathbb{F}_q)$. Then we get

$$e_n(g^s P, S) = e_n(g^{k+h(m)+n} P, g^l H(m)) = e_n(g^{h(m)+n} Q, H(m)).$$

Suppose now we have a couple (s, S), where $s \in \{1, \dots, \phi(n)\}$ and $S \in < P >$, such that

$$e_n(g^s P, S) = e_n(g^{h(m)+n} Q, H(m)).$$

Since $H(m), S \in < P >$, there are $u, v \in \{0, \dots, n-1\}$ such that $S = uP$ and $H(m) = vP$. Thus we get

$$e_n((g^s u - g^{h(m)+n+a} v) P, P) = 1.$$

The element $e_n(P, P)$ is a primitive nth root of 1 and so, we obtain

$$uv^{-1} \equiv g^{a+h(m)+n-s} \pmod{n},$$

Putting $l = a + h(m) + n - s \bmod \phi(n)$ and $k = a - l \bmod \phi(n)$, we get

$$s = k + h(m) + n \bmod \phi(n) \quad \text{and} \quad S = g^l H(m).$$

It follows that $(s, x(S), b)$ is the signature of m (where b is a bit determining S).

3 The Elliptic Curve and the Pairing

In this section we show how we can construct an elliptic with the desired properties in order to implement our signature scheme. This task is achieved by the following algorithm:

1. select two large prime numbers p_1 and p_2 such that the factorization of $p_1 - 1, p_2 - 1$ are known and the computation of the factorization of $n = p_1 p_2$ is unfeasible;
2. select a random prime number p and compute $m = \mathrm{ord}_n(p)$;
3. find, using the algorithm of [4], a supersingular elliptic curve E over $\mathbb{F}_{p^{2m}}$ with trace $t = 2p^m$;
4. return $\mathbb{F}_{p^{2m}}$ and E.

Since the trace of E is $t = 2p^m$, we get $|E(\mathbb{F}_{p^{2m}})| = (p^m - 1)^2$. On the other hand, we have $m = \mathrm{ord}_n(p)$, whence $n | p^m - 1$, and so n is a divisor of $|E(\mathbb{F}_{p^{2m}})|$. Therefore $E(\mathbb{F}_{p^{2m}})$ contains a subgroup of order n.

By Bróker [4, Theorem 1.1], we obtain, under the assumption that the Generalized Riemman Hypothesis is true, that the time complexity of Step 3 is $\tilde{O}((\log p^{2m})^3)$. Furthermore, since the factorization of $\phi(n) = (p_1 - 1)(p_2 - 1)$ is known, the time needed for the computation of m is $O((\log n)^2 / \log\log n)$ [15, Section 4.4].

For the implementation of our signature scheme we also need a point P with order n and an efficiently computable pairing e_n such that $e_n(P, P)$ is a primitive nth root of 1. The Weil pairing does not fulfill this requirement and also, in many instances, the Tate pairing; the same happens for the eta pairing (the eta and omega pairings can be computed only on the ordinary elliptic curves) [1, 10, 28]. Let ϵ_n be one of the previous pairings on $E[n]$. Following the method introduced by Verheul [23], we use a distortion map ϕ such that the points P and $\phi(P)$ is a generating set for $E[n]$ and we consider the pairing $e_n(P, Q) = \epsilon_n(P, \phi(Q))$. The algorithm of [7, Section 6] provides us a method for the determination of P and ϕ.

Another method for the construction of the elliptic curve E which is quite efficient in practice is given by the following algorithm:

1. draw at random a prime number p_1 of a given size l (for example, l is 1024 bits);
2. draw at random a number p_2 of size l;
3. repeat $p_2 = \mathrm{NextPrime}(p_2)$ until $4p_1p_2 - 1$ is prime;
4. return $p = 4p_1p_2 - 1$.

It is not proved that this algorithm will stop with a large probability. This is an open problem which is for $p_1 = 2$ the Sophie Germain number problem. But in practice we obtain a result p which is a prime of length $2l$.

Since $p \equiv 3 \pmod 4$, the elliptic curve defined over $\mathbb{F}_p$ by the equation

$$y^2 = x^3 + ax,$$

where $-a$ is not a square in $\mathbb{F}_p$, is supersingular with $p+1 = 4p_1p_2$ points. By Vladut [26, Theorem 2.1], the group $E(\mathbb{F}_p)$ is either cyclic or $E(\mathbb{F}_p) \simeq \mathbb{Z}/2p_1p_2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$. In each case the group $E(\mathbb{F}_p)$ has only one subgroup of order $n = p_1p_2$, and this subgroup is cyclic.

If ϵ_n is one of the Weil, Tate, or eta pairings on $E[n]$, then we use the distortion map $\phi(Q) = \phi(x, y) = (-x, iy)$ with $i^2 = -1$ (cf. [14]) and so, we obtain the following pairing: $e_n(P, Q) = \epsilon_n(P, \phi(Q))$.

4 The Map to Point Function

Let G be the subgroup of order $n = p_1p_2$ of $E(\mathbb{F}_q)$ introduced in the previous section. In order to sign using the discrete logarithm problem on this group, we have to define a hash function into the group G, namely a map to point function. This problem was studied by various authors giving their own method, for example in [3] or [11]. We give here the following solution. Let us denote by $|n| = \lfloor \log_2(n) \rfloor + 1$ the size of n. Let h be a key derivation function, possibly built using a standard hash function. We recall that h maps a message M and a bitlength l to a bit string $h(M, l)$ of length l. Moreover we will suppose that h acts as a good pseudo-random generator. Let Q be a generator of the group G. Let us denote by $(T_i)_{i \geq 0}$ the sequence of bit strings defined by $T_0 = 0$ and for $i \geq 1$

$$T_i = a_u \cdots a_0,$$

where $i = \sum_{j=0}^{u} a_j 2^j$ and $a_u = 1$.

To map the message m to a point $H(m)$ we run the following algorithm:

```
  $i := 0$;
Repeat
$k := h(m||T_i, |n|)$;
$i := i + 1$;
Until $k < n$;
Output $H(M) = k.Q$;
```

This Las Vegas algorithm has a probability zero to never stop. In practice this algorithm stops quickly, namely as $2^{|n|-1} < n < 2^{|n|}$ then the expected value of the number of iterations is <2. If one can find a collision for H it is easy to find a collision for h.

5 Performance Analysis

In this section we analyze the performance of our scheme. The computation of s requires two additions modulo $\phi(n)$. The computation of S needs a modular exponentiation $g^l \pmod{n}$ and the computations of $H(m)$ and $g^l H(m)$. Note that the computation of $g^l \bmod n$ and $k + n \bmod \phi(n)$ can be done off-line. Thus, the signature generation requires only a modular addition and a point multiplication on the elliptic curve. The signature verification needs two modular exponentiations, two points multiplications on the elliptic curves, and two pairing computations. Moreover note that the length of the signature of a message is the double of its length.

The signature generation in the GPS scheme [8] needs only one modular exponentiation and the signature verification two. The signature length is the triple of the message length. The most efficient of the schemes given in [12, 13, 19, 24, 25, 27] requires three modular exponentiations for the signature generation and four modular exponentiations for the signature verification. The signature length of the above schemes is larger than the double of the message length.

Hence we see that the signature length in our scheme is smaller than that in GPS and the other schemes. Moreover, the performance of the proposed algorithm is competitive to the performance of the above schemes.

6 Example

In this section we give an example of our signature scheme. We consider the 1024-bits primes

$$\begin{aligned} p_1 := {} & 6108796057503878981698853611415079226637763635184317758756 4\\ & 3192462711995704175406099915839974976783389653390629685931 1\\ & 2548516341523155127521258304405215057761482861700580373038 9\\ & 4387740068924296027884510970369084302618887384791344223443 2\\ & 3659125568423449336215957210074769940424533921400807874383 6\\ & 7162669180839 \end{aligned}$$

and

$$\begin{aligned} p_2 := {} & 95079457578903619398528949410023827176491364934193644644108 1\\ & 37707250057803575453826890251814298296023405531971834817156 4\\ & 53183534801316967559857543439452826972912632712819071175819 3\\ & 48708839569650309030711130343387015511459961721710564804000 5\\ & 34450679689842289797748919611061026066566455365600107406808 7\\ & 13249343. \end{aligned}$$

We take $n = p_1p_2$. The number $q = 4n - 1$ is a prime. Since $q \equiv 3 \pmod 4$, the elliptic curve E defined by the equation $y^2 = x^3 + x$ over $\mathbb{F}_q$ is supersingular. The point $P = (x(P), y(P))$, where $x(P) = 2^{1500} + 2$ and

$$y(P) = 92629334720096485394250229023531473128561210303747369871170$$

$$\begin{gathered}
532503591346084781038053790347765721405539373837575715741111302632\\
222520728502603977901582753916707479492439228918725855423715991340\\
003621514555505206507732534242013847767107764800751435936328543137\\
789247911179152023276247696951339536945505339588067200491193957998\\
044975563046555194785086909103272771864842171753848435480722850484\\
547366650914307823107502201128733622163636510656608071825566283432\\
994640380462713709910638633429178083083878848700277309884412794341\\
026781057881112432733889255328105052291841518470922081921433382412\\
47201267812054612564072614 8962.
\end{gathered}$$

has order n. We take $g = 2$,

$$\begin{gathered}
a = 2^{256} + 2^{9} + 1 = 115792089237316195423570985008687907853269984665\\
640564039457584007913129640449
\end{gathered}$$

and we compute

$$\begin{gathered}
g^{a} \bmod n = 291246612437704212466554616370488460582482345\\
412043139387071627568366461190658309237330580043030838224854789252\\
968050905018578440545530480131761225347896913705349073419345335895\\
868832920014327349522957752032149784650672578527400186028060209053\\
035728070430079944852013985987562947197675511448867860271390438151\\
997510376157277527652722786834963496843487625119512000324307142997\\
876216044005309541179123902262183075125684914484636806915549910481\\
194533920018176890664864601123368083711476432553316859751469426810\\
204461407620204756483516542976417259702626996120442929825569733396\\
712622105195095244311593920926256171476744 3.
\end{gathered}$$

Next, we compute $Q = g^{a}P = (x(Q), y(Q))$, where

$$\begin{gathered}
x(Q) = 4929066269630890940118676840165480358358027921633777075970 56\\
795455537761970341320418289803336076175870732053896841006011789243\\
411173491601076264818884432777686675649566399360544060115589059409\\
495626348669253033853643920668587107209662122339196308521380419432
\end{gathered}$$

$$\begin{gathered}
3958767770010377591298098261888264447928963024835312975003285776611 \\
1156441376633776947815847988008319196552077880554266338219162536488 \\
5455422641818199238687159366040776610195158709092926451452926125822 \\
0820564544916736264069574112504476158054648006035374272664210840677 \\
0688899424879273678267062426009254707550914157923366582588873582333 \\
664801117316512758157989323
\end{gathered}$$

and

$$\begin{gathered}
y(Q) = 9251640006679849414362134638435628671328426925266395037136233 \\
1007610587593256539123868607426378281972116750233717652921901662255 \\
6889076587632786360429521239281996051884310217309505235221721760611 \\
2499163363529422455175409284709873271636908991699714235667300461466 \\
0401314617119825149525737613057257718590923730935907182295497757288 \\
3180913934597216850220500675730525413684644075563296631876920873255 \\
7853188066562736344515028989009339090827154585880138328472819829188 \\
0452504062174178921959822834145697232804630292818810258440117103133 \\
0036374232447169484309288773766481841241697043304934210730109599044 \\
200046895734399896253588694.
\end{gathered}$$

Therefore $(P, Q, 2, n)$ and (a, p_1, p_2) are a public key and the corresponding private key for our signature scheme. Moreover, we can use the Tate pairing with the distortion map $\phi(x, y) = (-x, iy)$ with $i^2 = -1$.

7 Security of the Scheme

In this section we shall discuss the security of our system. First, we remark that if an attacker wants to compute the private key (a, p_1, p_2) from the public key, he has to factorize n and to compute the discrete logarithm g^a of Q to the base P and next to calculate the discrete logarithm a of g^a to the base g in the group $\mathbb{Z}_n$. Note that an algorithm which computes the discrete logarithm modulo n implies an algorithm which breaks the Composite Diffie–Hellman key distribution scheme for n and any algorithm which breaks this scheme for a non-negligible proportion of the possible inputs can be used to factorize n [2, 21].

In order to study the security of the scheme we are going to look at the two worst cases:

1. the factorization problem is broken but the elliptic curve discrete logarithm problem is not;
2. the elliptic curve discrete logarithm problem is broken but the factorization problem is not.

In each case we will prove that if an attacker is able to generate a valid signature for any given message m, then it is able to solve, in the first case the elliptic curve discrete logarithm problem and in the second case the factorization problem.

1. Let us suppose that the attacker is able to factorize n. Then he can compute $\phi(n)$. But he is unable to compute a since a is protected by the elliptic curve discrete logarithm problem and by the discrete logarithm problem modulo n, because the only known relation involving a is $Q = g^a P$. So, in order to produce a valid signature of a message m the attacker has only two possibilities: he can arbitrary choose k, and then he can compute s but not S, or choose arbitrary l and he can compute S but not s.
2. Let us suppose now that the attacker is able to solve the elliptic curve discrete logarithm problem. Then he can compute g^a but as the factorization problem is not broken the discrete logarithm problem modulo n is not broken and consequently he cannot compute a (cf. the beginning of this section). Then as in (1) he cannot compute simultaneously s and S.

8 Conclusion

In this paper we defined a signature system based on two difficult arithmetic problems. In the framework chosen, these problems have similar resistance to known attacks. We explained how to implement in practice all the basic functions we need for the establishment and operation of this system. This strategy has an interest in any application that includes a signature to be valid for long. Indeed, it is hoped that if any of the underlying problems is broken, the other will still be valid. In this case, the signature should be regenerated with a new system, without the chain of valid signatures being broken. Finally, the signature length of our scheme is smaller than that of the schemes based on integer factorization and integer discrete logarithm problems, and its performance is competitive to that of these schemes.

References

1. Barreto, P.S.L., Galbraith, S.D., Ó'hÉigeartaigh, C., Scott, M.: Efficient pairing computation on supersingular Abelian varieties. Des. Codes Crypt. **42**, 239–271 (2007)
2. Biham, E., Boneh, D., Reingold, O.: Breaking generalized Diffie-Hellman is no easier than factoring. Inf. Proces. Lett. **70**, 83–87 (1999)

3. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. Lect. Notes Comput. Sci. **2248**, 514–532 (2001)
4. Bróker, R.: Constructing supersingular elliptic curves. J. Combin. Number Theory **1**(3), 269–273 (2009)
5. Buchmann, J., May, A., Vollmer, U.: Perspectives for cryptographic long term security. Commun. ACM **49**(9), 50–55 (2006)
6. Chen, T.-H., Lee, W.-B., Horng, G.: Remarks on some signature schemes based on factoring and discrete logarithms. Appl. Math. Comput. **169**, 1070–1075 (2005)
7. Galbraith, S.D., Rotger, V.: Easy decision Diffie-Hellman groups. LMS J. Comput. Math. **7**, 201–218 (2004)
8. Girault, M., Poupard, G., Stern, J.: Global Payment System (GPS): un protocole de signature à la volée. In: Proceedings of Trusting Electronic Trade, 7–9 June 1999
9. Harn, L.: Enhancing the security of ElGamal signature scheme. IEE Proc. Comput. Digital **142**(5), 376 (1995)
10. Hess, F., Smart, N.P., Vercauteren, F.: The Eta pairing revisited. IEEE Trans. Inf. Theory **52**(10), 4595–4602 (2006)
11. Icart, T.: How to Hash into elliptic curves. In: CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 303–316. Springer, New York (2009)
12. Ismail, E.S., Tahat, N.M.F., Ahmad, R.R.: A new digital signature scheme based on factoring and discrete logarithms. J. Math. Stat. **4**(4), 22–225 (2008)
13. Ismail, E.S., Tahat, N.M.F.: A new signature scheme based on multiple hard number theoretic problems. ISRN Commun. Netw. **2011**, 3 pp. (2011). Article ID 231649. http://dx.doi.org/10.5402/2011/231649
14. Joux, A.: The weil and tate pairings as building blocks for public key cryptosystems (Survey). In: ANTS 2002. Lecture Notes in Computer Science, vol. 2369, pp. 20–32. Springer, Berlin (2001)
15. Karagiorgos, G., Poulakis, D.: Efficient algorithms for the basis of finite Abelian groups. Discret. Math. Algoritm. Appl. **3**(4), 537–552 (2011)
16. Lee, N.Y.: Security of Shao's signature schemes based on factoring and discrete logarithms. IEE Proc. Comput. Digital Tech. **146**(2), 119–121 (1999)
17. Lee, N.Y., Hwang, T.: The security of He and Kiesler's signature scheme. IEE Proc. Comput. Digital Tech. **142**(5), 370–372 (1995)
18. Li, J., Xiao, G.: Remarks on new signature scheme based on two hard problems. Electron. Lett. **34**(25), 2401 (1998)
19. Madhur, K., Yadav, J.S., Vijay, A.: Modified ElGamal over RSA Digital Signature Algorithm (MERDSA). Int. J. Adv. Res. Comput. Sci. Softw. Eng. **2**(8), 289–293 (2012)
20. Maseberg, J.S.: Fail-safe konzept fur public-key infrastrukturen. Thesis, Technische Universitat Darmstadt (2002)
21. McCurley, K.S.: A key distibution system equivalent to factoring. J. Cryptol. **1**, 95–105 (1988)
22. Shao, Z.: Security of a new digital signature scheme based on factoring and discrete logarithms. Int. J. Comput. Math. **82**(10), 1215–1219 (2005)
23. Verheul, E.: Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. In: Advances in Cryptology-Eurocrypt '01. Lecture Notes in Computer Science, vol. 2045, pp. 195–210. Springer, New York (2001)
24. Verma, S., Sharma, B.K.: A new digital signature scheme based on two hard problems. Int. J. Pure Appl. Sci. Technol. **5**(2), 55–59 (2011)
25. Vishnoi, S., Shrivastava, V.: A new digital signature algorithm based on factorization and discrete logarithm problem. Int. J. Comput. Trends Tech. **3**(4), 653–657 (2012)
26. Vladut, S.: Cyclicity statistics for elliptic curves over finite fields. Finite Fields Appl. **5**(1), 13–25 (1999)
27. Wei, S.: A new digital signature scheme based on factoring and discrete logarithms. In: Progress on Cryptography. Kluwer International Series in Engineering and Computer Science, vol. 769, pp. 107–111. Kluwer Academic Publishers, Boston (2004)
28. Zhao, C.-A., Xie, D., Zhang, F., Zhang, J., Chen, B.-L.: Computing bilinear pairing on elliptic curves with automorphisms. Des. Codes Crypt. **58**, 35–44 (2011)