Searching BN Curves for SM9

Guiwen Luo[1,2(✉)] and Xiao Chen[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{luoguiwen,chenxiao}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. In 2016, State Cryptography Administration of China published Identity-based cryptographic algorithm SM9. A 256-bit BN curve recommended to construct system parameters in SM9 documents once was convinced to provide 128-bit security level. With the development of number field sieve, the complexity of discrete logarithm problem (DLP) in a finite field reduces, so does the security level of SM9 whose security is based on the difficulty of solving the DLPs. It's urgent to construct SM9 system parameters with higher security level. In this paper, we analyze the requirements of secure elliptic curves, search BN curves at length of 384-bit and 380–382-bit that show the best computation efficiency. Then we choose a 384-bit BN curve to construct the system parameters, making preparation for upgrading the original 256-bit SM9.

Keywords: SM9 · Identity-based cryptographic algorithm · System parameters · BN curve · R-ate pairing

1 Introduction

SM9 is an identity-based cryptosystem [1–5] based on bilinear pairing which makes connections between the cyclic subgroups of elliptic curves and the cyclic multiplication subgroup of finite field. SM9 chooses a BN curve to construct its system parameters and R-ate Pairing to implement all the cryptographic algorithms.

From the perspective of mathematics, SM9 is also a pairing-based cryptosystem whose security is based on the difficulty of DLPs on elliptic curves and finite field. Original SM9 has recommended 256-bit system parameters in order to provide 128-bit security level at the time. In 2016, there was a big improvement on number field sieve (NFS) algorithm [18], bringing down the difficulty of DLP in finite field. A variant of ordinary NFS named special ExTNFS is suitable for extension field $\mathbb{F}_{q^n}$ with prime q given by a polynomial of parameter t, such as the cases of BN curves. This leads to a result that the 256-bit BN curve in original SM9 could theoretically just provide the security level of around 100-bit [6]. It's urgent to construct system parameters of SM9 with higher security level to make preparation for upgrade.

F. Guo et al. (Eds.): Inscrypt 2018, LNCS 11449, pp. 554–567, 2019.
https://doi.org/10.1007/978-3-030-14234-6_29

Several recent works have been done to revise the key size of corresponding security level. Menezes et al. [20], Barbulescu et al. [6], and Scott et al. [28] proposed new key size estimations for pairing-based cryptography. Pairing-friendly elliptic curves, among which the most popular ones are BN curves [9], are commonly utilized to construct system parameters for pairing-based cryptosystem. A general survey about different families of pairing-friendly elliptic curves and their constructions is introduced in [14].

In this article, conditions that secure curves need to satisfy are analyzed and why those conditions are necessary is explained. An exhaustive search over Hamming weight of curve parameter is made, obtaining the best secure BN curves at length of 384-bit and 382-bit respectively. Then we compare and combine state-of-the-art algorithms to evaluate the R-ate pairing computation efficiency, which is the core part of SM9. Finally, the 384-bit BN curve is chosen to construct SM9 system parameters. In a word, we've done all the essential work to upgrade 256-bit SM9 to a higher security level.

2 Preliminaries

2.1 BN Curves

BN curves are a very important family of pairing-friendly elliptic curves broadly used in pairing-based cryptography. A BN curve is represented by integer triplet (t, q, n) with the relationship

$$\begin{aligned} q &= q(t) = 36t^4 + 36t^3 + 24t^2 + 6t + 1, \\ n &= n(t) = 36t^4 + 36t^3 + 18t^2 + 6t + 1, \end{aligned} \tag{1}$$

where both q and n are prime integers. A 256-bit BN curve means the binary length of q and n is 256-bit. The curve equation can be written as

$$E : y^2 = x^3 + b,\ b \in \mathbb{F}_q. \tag{2}$$

The embedding degree of BN curve is $k = 12$, making it appropriate for pairing-based cryptography. Another important property of BN curves is that a twisted curve with degree 6 exists [24]. The twist is represented by the equation

$$E' : y^2 = x^3 + b/\beta, \tag{3}$$

where $\beta \in \mathbb{F}_{q^2} \setminus ((\mathbb{F}_{q^2})^2 \cup (\mathbb{F}_{q^2})^3)$. It helps us to represent the second R-ate pairing argument point in a quadratic extension field to achieve higher pairing computation efficiency. The corresponding isomorphism $\psi \in \hom(E', E)$ is

$$\psi : E' \to E, (x', y') \mapsto (\beta^{1/3}x', \beta^{1/2}y'). \tag{4}$$

2.2 Bilinear Pairing

Let $(G_1, +), (G_2, +)$ and $(G_T, \cdot)$ be three cyclic groups with the same order of prime integer n. Let P_1 and P_2 be the generators of G_1 and G_2 respectively. Suppose there exists a homomorphism ψ such that $\psi(P_2) = P_1$.

A bilinear pairing is a map

$$e : G_1 \times G_2 \to G_T$$

with bilinearity, non-degeneracy and computability.

Different kinds of bilinear pairings, including Weil Pairing, Tate Pairing, Ate Pairing and R-ate Pairing, are suitable for SM9. Original 256-bit SM9 [5] chooses R-ate Pairing as the recommended one.

Define $G_1 = \ker(\phi_q - [1]) = E(\mathbb{F}_q)$ and $G_2 = E[n] \cap \ker(\phi_q - [q]) \subseteq E(\mathbb{F}_{p^{12}})[n]$, where ϕ_q is the Frobenius endomorphism and G_T is a subgroup of $\mathbb{F}_{q^{12}}$ with a prime order n. Suppose the straight line acrossing points U and V on a BN curve is $\lambda x + \delta y + \tau = 0$, let function $g_{U,V}(Q)$ be

$$g_{U,V}(Q) = \lambda x_Q + \delta y_Q + \tau, \ \textit{where}\ Q = (x_Q, y_Q).$$

If $U = V$, let $g_{U,V}$ be the tangent line acrossing U; if one of U and V is infinity O, let $g_{U,V}$ be the vertical line acrossing the other point. Then R-ate pairing on a BN curve can be computed as Algorithm 1 [1], which contains three key steps—the Miller loop (Line 1–10), adjustment step (Line 11–16) and final exponentiation (Line 17).

2.3 Algorithm Attacks on SM9

The security of SM9 is determined by the difficulty of discrete logarithm problem (DLP) in G_1, G_2, G_T. There are two different kinds of DLPs, one is the DLP in elliptic curves, the other is the DLP in finite field, i.e. G_T. The cost of DLPs is required to be big enough to meet the corresponding security level. This subsection introduces the most effective algorithm attacks on SM9. Those attacks are suitable for pairing-based cryptography, too.

The effective attack algorithm in curve side G_1, G_2 is Pollard's rho algorithm [14,25]. Since we employ BN curves in SM9, $q(t)$ and $n(t)$ have the same bit length, that we denote as l_1. Security evaluation on G_1 and G_2 is simple, $l_1 \geq 2l_2$ is sufficient for the requirement of l_2-bit security level.

Algorithm 1. Computing R-ate pairing on BN Curves

Input:$P \in G_1, Q \in G_2, a = |6t + 2|$.

Output:R-ate pairing f.

Let $a = \sum_{i=0}^{L-1} a_i 2^i, a_{L-1} = 1, a_i \in \{-1, 0, 1\}$.
$T \leftarrow Q, f \leftarrow 1$.
for $i = L - 2$ *to* 0 **do**
  $f \leftarrow f^2 \cdot g_{T,T}(P), T \leftarrow [2]T$;
  if $a_i = 1$ **then**
    $f \leftarrow f \cdot g_{T,Q}(P), T \leftarrow T + Q$.
  elseif $a_i = -1$ **then**
    $f \leftarrow f \cdot g_{T,-Q}(P), T \leftarrow T - Q$.
  endif
endfor
if $t < 0$ **then**
$T \leftarrow -T,\ f \leftarrow f^{q^6}$.
endif
$Q_1 \leftarrow \phi_q(Q), Q_2 \leftarrow \phi_{q^2}(Q)$.
$f \leftarrow f \cdot g_{T,Q_1}(P), T \leftarrow T + Q_1$.
$f \leftarrow f \cdot g_{T,-Q_2}(P)$.
$f \leftarrow f^{(q^{12}-1)/n}$.
return f.

Another attack in curve side is advised by Cheon [11,12]. Cheon shows that the strong Diffie-Hellman (SDH) problem with auxiliary inputs can be solved faster than ordinary DLP. Suppose attacker can repeat signatures for k times and collect the corresponding public keys, if $n - 1$ contains a divisor $d \leq \min\{k + 1, n^{1/2}\}$, or $n + 1$ contains a divisor $d \leq \min\{(k + 1)/2, n^{1/3}\}$, then the secret key can be found in $O(\sqrt{n/d})$, in other words security level could be reduced by $O(\sqrt{d})$. This implies that the security level of SDH problem could be lower than we expect if system parameters are randomly chosen. It is recommended to select a secure curve with a prime order n such that both $n + 1$ and $n - 1$ have no small divisor.

The best attack algorithm in finite field side is the number field sieve (NFS) algorithm. A variant of NFS, named special extended tower-NFS [7,18] (Special ExTNFS), is dedicated to extension field $\mathbb{F}_{q^n}$ with prime q given by a polynomial of parameter t, such as the case of BN curves. Since this area is developing, we can't give a clear and precise security evaluation in finite field side. Theoretical improvements on special ExTNFS have been done but real-life implementations are not available. In particular the relation collection step is a tough work and is not implemented at present. But we should still take those theoretical improvements seriously when evaluating the security level.

3 Conditions of SM9 Secure Curves

System parameters are carefully selected to ensure that SM9 system runs securely. They includes the elliptic curve parameters, the curve identifier, the order of cyclic group and it's cofactor, etc. The most important one is the elliptic curve which meets all the security requirements of SM9. Such elliptic curves are called secure curves. Since the supersingular curves are proved to be insecure, we focus on ordinary curves. Part 1 of SM9 documents states three conditions that a secure curve needs to meet:

Condition 1. Ordinary curve whose base field is $\mathbb{F}_q$, where q is a prime number greater than 2^{191}. The embedding degree $k = 2^i \cdot 3^j$, where $i > 0, j >= 0$.

Condition 2. $n - 1$ contains a prime factor greater than 2^{190}.

Condition 3. $n + 1$ contains a prime factor greater than 2^{120}.

The **Condition 2** and **Condition 3** are specified to reduce the impact caused by Cheon Attack. It's more appropriate to modify those two conditions as containing small prime factors as few as possible.

The 256-bit BN curve, presented in the SM9 documents, once was considered that the G_T provided security level higher than 128-bit, and that the weakness of the system was G_1 and G_2, whose security levels are no more than 128-bit by Pollard's rho algorithm. It was necessary to reduce the impact on G_1 and G_2 caused by Cheon Attack at the time. With the development of special ExTNFS, it's now commonly accepted that curve side is stronger than finite field side, so the urgency of those two conditions has gone, they can be the last to be considered. Furthermore, $n-1$ and $n+1$ always contain small prime factor such as 2, and in real life Cheon Attack is limited by the amount of public keys the attacker can collect and by the total number of system identities.

Other conditions that need to be considered are

Condition 4. $2q - n$ is a prime number.

This condition guarantees subgroup security [8]. The 256-bit BN in SM9 doesn't take it into consideration, so membership test is required every time. For a BN curve, the order of $E(\mathbb{F}_q)$ is a prime number, it naturally protects against the subgroup attacks that exploit small prime divisors of the cofactor. However, this is not the case of $E'(\mathbb{F}_{q^2})$ since its order equals $n(2q - n)$. The subgroup attack under this circumstance can be prevented by using membership tests, which may be expensive. If we want to avoid these tests, the curve parameter t should be chosen such that both n and $2q - n$ are prime numbers.

Condition 5. the Hamming weight of BN curve parameter t should be as small as possible.

It obviously helps Miller loop for R-ate pairing computation, and it's also beneficial to final exponentiation.

Condition 6. $t = 2$ or 10 modulo 12.

This condition guarantees the tower extension of $\mathbb{F}_q$ to be the same as SM9-Part5, which means the reduction modulo polynomials of field extension are irreducible polynomials $x^i + 2,\ i = 2, 4, 6, 12 \in \mathbb{F}_q[x]$. Under such a condition we can partly reutilize software implementation of $\mathbb{F}_{q^{12}}$ in the original 256-bit SM9 to reduce the expenditure of updating SM9. Actually we've also considered another tower extension case with irreducible polynomials $x^2 + 1 \in \mathbb{F}_q[x]$ and

$x^6 - (1 + \sqrt{-1}) \in \mathbb{F}_{q^2}[x]$. This extension case is a little faster than the former one if the Hamming weight of their curve parameter ts are equal, but no t with Hamming weight no more than 6 meets the case.

4 Searching BN Curves for SM9

The curve parameter t completely decides a BN curve. We represent t in non-adjacent form (NAF), restrict $q(t)$ and $n(t)$ in Eq. (1) as 384-bit prime numbers, then exhaustively search on t with increase of Hamming weight. When Hamming weight is no more than 5, there is no such t that meets all the conditions listed in Sect. 3. When Hamming weight is 6, there are 5 ts (listed in Table 1) meet all the conditions.

Table 1. ts that make $(t, q(t), n(t))$ satisfy all the conditions listed in Sect. 3. p_{n+1} denotes the biggest prime factor of $n + 1$. p_{n-1} denotes the biggest prime factor of $n - 1$.

NAF t	Length of q and n	$W_H(6t + 2)$	$Len(p_{n+1})$	$Len(p_{n-1})$
$-2^{95} + 2^{93} + 2^{61} + 2^{57} - 2^{31} + 2$	384	10	173	283
$-2^{95} + 2^{93} - 2^{63} - 2^{35} - 2^{32} - 2$	384	9	167	287
$-2^{95} + 2^{93} + 2^{80} - 2^{71} - 2^{58} - 2$	384	10	158	250
$2^{95} - 2^{93} + 2^{85} + 2^{31} - 2^3 + 2$	384	8	127	234
$-2^{95} + 2^{93} - 2^{91} - 2^{67} - 2^{65} + 2$	384	7	144	195

Although we aim to search 384-bit BN curves, but in practice the case when BN curves are 380–382-bit is also attractive. The binary length of such parameters aren't precisely the multiple of word size. It shows unique advantages when computing R-ate pairing on processors of 32-bit or 64-bit. We can employ residue number systems (RNS) [19] and lazy reduction techniques to improve the computation efficiency. Lazy reduction is well suited for expressions like $AB \pm CD \in \mathbb{F}_q$. According to normal steps, 2 module reductions are required, while lazy reduction performs only 1 module reduction. RNS and lazy reduction can be employed effectively only when the size of q is chosen to be a little bit smaller than an exact multiple of the word size of the processor architecture. When Hamming weight is no more than 4, no such t meets all the conditions listed in Sect. 3. When t's Hamming weight is 5, there are 2 ts (listed in Table 2) meet all the conditions.

Since the R-ate pairing computation efficiency is determined by the length of q, the Hamming weight of t and $6t + 2$, we can make further selection among those security curves. We denote the most efficient 384-bit BN security curve as t_{384}, and the 382-bit one as t_{382}, then

$$\begin{aligned} t_{384} &= -2^{95} + 2^{93} - 2^{91} - 2^{67} - 2^{65} + 2, \\ t_{382} &= -2^{94} - 2^{81} - 2^{11} - 2^3 + 2. \end{aligned} \tag{5}$$

Table 2. ts that make $(t, q(t), n(t))$ satisfy all the conditions listed in Sect. 3. p_{n+1} denotes the biggest prime factor of $n+1$. p_{n-1} denotes the biggest prime factor of $n-1$.

NAF t	Length of q and n	$W_H(6t+2)$	$Len(p_{n+1})$	$Len(p_{n-1})$
$-2^{94}-2^{81}-2^{11}-2^3+2$	382	8	180	203
$-2^{94}-2^{89}-2^{62}-2^5-2$	382	10	172	215

5 R-ate Pairing Computation

In this section, we suppose the basic arithmetic operations (addition, subtraction, multiplication and inversion) in $\mathbb{F}_q$ have been implemented, which means our analysis is independent of the underlying hardware architecture. Those basic arithmetics usually employ schoolbook method or Karatsuba method [19] and the Montgomery reduction [21] or the Barrett reduction [10]. Another way to implement $\mathbb{F}_q$ is to use residue number system [19] and lazy reduction [26]. Whatever the algorithms are employed in $\mathbb{F}_q$, it doesn't influence our efficiency comparison when computing R-ate pairings using different curve parameter t. We set the notations for $\mathbb{F}_{q^i}$ $(i = 1, 2, 4, 6, 12)$ arithmetics as follows: A_i denotes an addition and A_i' denotes a doubling, M_i denotes a multiplication, sM_i denote a sparse multiplication (which is employed in Miller loop), S_i denotes a squaring and I_i denotes an inversion. We take the pairing computation down to the basic arithmetics in $\mathbb{F}_q$ to show the efficiency of pairing computation based on curve parameter t_{384} and t_{382} in Eq. (5).

5.1 Complexities of Arithmetics in Tower Extension Field

The well-known strategy to improve performance in $\mathbb{F}_{q^{12}}$ is to represent $\mathbb{F}_{q^{12}}$ using tower extension trick, it's faster than directly represent $\mathbb{F}_{q^{12}}$ as

$$\mathbb{F}_{q^{12}} = \mathbb{F}_q[w]/(w^{12}+2).$$

The most efficient tower extension from $\mathbb{F}_q$ to $\mathbb{F}_{q^{12}}$ is 2-3-2 extension [13]. We assume that additions are not negligible and $A_1 \leq 0.33M_1$, under this assumption multiplications in all extension level (i.e. in $\mathbb{F}_{q^k}$, $k = 2, 6, 12$) are implemented by Karatsuba arithmetic, and the squaring in the degree 3 extension $(\mathbb{F}_{q^6}/\mathbb{F}_{q^2})$ is implemented by Chung-Hasan method.

Since our curve parameter $t = 2$ or 10 modulo 12, we can use the same tower extension as the original 256-bit SM9. The tower extension of $\mathbb{F}_q$ is as below:

$$\begin{aligned}
\mathbb{F}_{q^2} &= \mathbb{F}_q[u]/(u^2-\alpha), \alpha = -2,\\
\mathbb{F}_{q^6} &= \mathbb{F}_{q^2}[v]/(v^3-u), u^2 = \alpha,\\
\mathbb{F}_{q^{12}} &= \mathbb{F}_{q^6}[w]/(w^2-v), v^3 = u.
\end{aligned}$$

Table 3. Complexities of arithmetics in tower extension fields

Operation	Number of operations
I_2	$I_1 + 2M_1 + 2S_1 + A_1 + A_1'$
M_2	$3M_1 + 5A_1 + A_1'$
S_2 (Complex method)	$2M_1 + 3A_1 + 2A_1'$
I_6	$I_1 + 35M_1 + 2S_1 + 65A_1 + 20A_1'$
M_6	$18M_1 + 60A_1 + 8A_1'$
S_6	$12M_1 + 35A_1 + 12A_1'$
I_{12}	$I_1 + 95M_1 + 2S_1 + 261A_1 + 61A_1'$
M_{12}	$54M_1 + 210A_1 + 25A_1'$
sM_{12}	$39M_1 + 115A_1 + 16A_1'$
S_{12}	$36M_1 + 129A_1 + 37A_1'$
S_{12} (Karabina's squaring)	$12M_1 + 50A_1 + 23A_1'$
S_{12} (Simult. decompression of k elt.)	$I_1 + (21k-7)M_1 + 2S_1 + (47k-14)A_1$ $+(21k-2)A_1'$
S_{12} (Granger and Scott's method)	$18M_1 + 75A_1 + 34A_1'$

We count the operations of tower extensions based on $\mathbb{F}_q$ in Table 3. Note that the cyclotomic squaring in $\mathbb{F}_{q^{12}}$, which is utilized in the final exponentiation, is assumed to be implemented by Karabina's compress method [17] and Montgomery's simultaneous inversion trick [22]. Another cyclotomic squaring algorithm is Granger and Scott's method [16], which in most case is slower than Karabina's method unless the $\mathbb{F}_{q^2}$ inversion is very time consuming.

5.2 Complexities of R-ate Pairing Computation

R-ate pairing computation efficiency is determined by the length of q, the Hamming weight of t and $6t+2$. We utilize projective coordinates for adding and doubling points along with line computation in Miller loop. Scott et al.'s method [27] is taken to implement the final exponentiation step. Although Fuentes-Castaneda et al. [15] provide another method which can save $3M_{12}+1S_{12}$, but their method demands the redefinition of R-ate pairing to its fixed power. We count the number of operations for R-ate pairing corresponding to t_{384} and t_{382}, the result is presented in Table 4. It seems that pairing computation corresponding to t_{382} is just slightly faster than that corresponding to t_{384}, but in practice when utilizing RNS and lazy reduction, the advantage could be greater.

6 Matching Security Level

The security of SM9 is determined by the difficulty of discrete logarithm problem in G_1, G_2 (curve side) and G_T (finite field side).

Table 4. Complexities of R-ate pairing computation corresponding to t_{384} and t_{382}.

Operation	t_{384}	t_{382}
Doubling+LineEval.	$24M_1 + 59A_1 + 16A_1'$	$24M_1 + 59A_1 + 16A_1'$
Addition+LineEval.	$44M_1 + 80A_1 + 16A_1'$	$44M_1 + 80A_1 + 16A_1'$
Miller loop	$10101M_1 + 30561A_1 + 6885A_1'$	$10184M_1 + 30756A_1 + 6917A_1'$
Adjustment steps	$170M_1 + 390A_1 + 64A_1'$	$170M_1 + 390A_1 + 64A_1'$
Final exponentiation	$4I_1 + 5644M_1 + 8S_1$ $+ 21915A_1 + 7874A_1'$	$4I_1 + 5383M_1 + 8S_1$ $+ 20994A_1 + 7667A_1'$
R-ate pairing	$4I_1 + 15915M_1 + 8S_1$ $+ 52866A_1 + 14823A_1'$	$4I_1 + 15737M_1 + 8S_1$ $+ 52140A_1 + 14648A_1'$

We analyze the BN curve corresponding to t_{384} first. The curve order n is a 384-bit prime number, so its security level is no more than 192-bit by Pollard's rho algorithm. Then we consider Cheon attack, since

$$\begin{aligned} n + 1 = \; & 2 \cdot 5 \cdot 7 \cdot 11 \cdot 131 \cdot 42992652371 \cdot 28839188139379 \\ & 46545034377697 \cdot 43111743941006841034336199136 7 \cdot d_1, \end{aligned}$$

where d_1 is a 144-bit prime number,

$$\begin{aligned} n - 1 = \; & 2^2 \cdot 3 \cdot 61 \cdot 547271 \cdot 29406309859180669069321 \cdot \\ & 938329529874120886260312 91 \cdot d_2, \end{aligned}$$

where d_2 is a 195-bit prime number. We take $d = \max\{2 \cdot 5 \cdot 7 \cdot 11 \cdot 131 \cdot 42992652371,\ 2^2 \cdot 3 \cdot 61 \cdot 547271\}$, according to Cheon attack, the security level can be reduced by 26-bit (the length of $\sqrt{d}$). Note that this reduction is just theoretical since in real life the attacker can't collect so enormous amount of public keys to conduct such an attack. Combine all together, the curve side can provide around 166-bit security level.

As for the finite field side, Freeman et al. [14] has made a well-known speculation about the exact sizes of q and q^k (where k denotes the embedding degree of the curve) required to match corresponding security level in 2010. They believed that one could achieve 128-bit security level by choosing a 256-bit prime number q and a proper embedding degree k such that the binary length of q^k was fell in the range 3000–5000, that was what the original SM9 had done. With the size of q^k 3072, it was convinced to provide 128-bit security level at the time.

But with the development of the special ExTNFS, security on the finite field side decreases, thus the security evaluation of pairing-based cryptography need to be reconsidered. The problem is complicated by the fact that the Special ExTNFS is developing, although theoretical improvements have been done but real-life implementations are missing. Barbulescu et al. proposed a new key size estimation for pairing-based cryptography [6], and Michael Scott followed their analysis to give his estimation on the security requirement for key size [28]. Although those estimation are rough and theoretical, it provide the new trend

of security reduction in finite field side. According to [6,23,28], since we choose the BN curve corresponding to t_{384} whose embedding degree $k = 12$, special ExTNFS makes the security level be $(130 - \delta)$-bit, where δ is not precisely known. δ is usually about a dozen, so the 384-bit BN curve can theoretically provide around 118-bit security level. Note that a 384-bit BN curve might not be able to meet the 128-bit security level [23].

As the analysis shows, at present the weakness is on the finite field side. There are more than 62-bit security level margin for Cheon attack in the curve side, thus it's unnecessary to consider Cheon attack.

To sum up, BN curve corresponding to t_{384} theoretically provides around 118-bit security level. Follow the same analysis, BN curve corresponding to t_{382} provides security level which is 1-bit lower than that of t_{384} (Table 5).

Table 5. Security level provided by BN curves corresponding to t_{384} and t_{382}.

Curve parameter	Pollard rho	Cheon	G_1, G_2 security level	G_T security level	Security level
t_{384}	192-bit	26-bit	166-bit	$(130 - \delta)$-bit	$(130 - \delta)$-bit
t_{382}	191-bit	51-bit	140-bit	$(129 - \delta)$-bit	$(129 - \delta)$-bit

7 Constructing SM9 System Parameters with the 384-Bit BN Curve

When updating SM9 to a higher security level, two curves with different length are available—one is the 384-bit BN curve corresponding to t_{384}, another is the 382-bit BN curve corresponding to t_{382}. The 382-bit BN curve corresponding to t_{382} shows better R-ate pairing computation efficiency by taking advantages of RNS and lazy reduction in the context of sacrificing 1-bit security level. The issue of which one should be adopted when updating SM9 needs further discussion. We prefer to select the 384-bit one, here are our general opinions. As a specialized standard, it is more appropriate for SM9 to choose system parameters with the length of the multiple of word length. Furthermore, R-ate pairing computation is fast enough to achieve good performance when implementing SM9 with the 384-bit BN curve on software level. In addition to that, RNS and lazy reduction can also apply to the 384-bit BN curve by employing an extra word or by utilizing word length longer than the standard (for example, 36-bit) when implementing SM9 on hardware level, narrowing the efficiency gap to the 382-bit curve.

We choose the BN curve corresponding to t_{384} to construct SM9 system parameters, detailed definitions are contained in **Appendix A**.

8 Conclusion

In this paper, we analyze the conditions that the secure curves in SM9 system parameters need to meet, search 384-bit and 380–382-bit BN curves that satisfy

all the conditions and study state-of-the-art algorithms to compute R-ate pairing. Then we select two BN curves, corresponding to curve parameters t_{384} and t_{382} in Eq. (5), to analyze their R-ate pairing computation complexity and security level. Those two BN curves show best computation efficiency at the length of 384-bit and 382-bit respectively among security curves. Finally, we construct SM9 parameters with the 384-bit BN curve corresponding to t_{384}, finishing the core work of updating SM9. While the original 256-bit system parameters presented in SM9-Part5 provide around 100-bit security level, new system parameters proposed by this paper provide around 118-bit security level. Our methods are also suitable for constructing and analyzing BN curves at other binary length for SM9.

Acknowledgment. We'd like to thank Ning Ma, Baofeng Wu and Yalan Ma for discussions and proofreading. We also thank the anonymous reviewers for their helpful comments. This work is supported by the National Defense Science and Technology Innovation Foundation (No. Y7H0041102).

A Definitions of 384-bit SM9 System Parameters

We choose the BN curve with

$$t = -2^{95} + 2^{93} - 2^{91} - 2^{67} - 2^{65} + 2,$$

to construct 384-bit SM9 system, in this case

$$\begin{aligned} n + 1 = \; & 2 \cdot 5 \cdot 7 \cdot 11 \cdot 131 \cdot 42992652371 \cdot 28839188139379 \\ & 46545034377697 \cdot 43111743941006841034336199136 7 \cdot d_1, \end{aligned}$$

where d_1 is a 144-bit prime number,

$$\begin{aligned} n - 1 = 2^2 \cdot 3 \cdot 61 \cdot 547271 \cdot 29406309859180669069321 \cdot \\ 93832952987412088626031291 \cdot d_2, \end{aligned}$$

where d_2 is a 195-bit prime number.

Parameters are represented in hexadecimal. We use column vector to express element in extension field, with higher dimension above and lower dimension below. The tower extension is the same as Subsect. 5.1.

Elliptic Curve Equation: $y^2 = x^3 + b$.
Parameter of Curve t: `-68000009FFFFFFFFFFFFFFFE`
Trace $tr(t) = 6t^2 + 1$:

`FD800030C0000257FFFFFFF63FFFFF100000000000000019`

Characteristic of Ground Field $q(t) = 36t^4 + 36t^3 + 24t^2 + 6t + 1$:

`FB0640608C400DECD800E46E46DD77FBD1FF65C07FFB0F16`
`3400230A0001AF3FFFFFFD530FFFBE2400000000000003CD`

Equation Parameter b : 02
Order of BN Curve $E(\mathbb{F}_q)$ $n(t) = 36t^4 + 36t^3 + 18t^2 + 6t + 1$:

FB0640608C400DECD800E46E46DD77FBD1FF65C07FFB0F15
368022D94001ACE7FFFFFD5CCFFFBF1400000000000003B5

Cofactor cf : 1
Embedding Degree k : 12
Twisted Curve $(E'/\mathbb{F}_{q^2} : y^2 = x^3 + b/\beta)$ Parameter $\beta : \sqrt{-2}$
Curve ID cid : 0x12
Generator of Group G_1 $P_1 = (x_{P_1}, y_{P_1})$:
Coordinate x_{P_1} :

5DE44C2E23720EBADC3046A8579979ACCF7C98875AE0EE84
76408737A19B77F54C6DC206EF3D4466B71500FEE1E4E456

Coordinate y_{P_1} :

6AD86724D049835A067B8AC1AD42EF44FCBAD8FF9CA0EACC
2FCABB12B666492A69BAE4F0E6A87C650FBEAE0C0B579BF7

Generator of Group G_2 $P_2 = (x_{P_2}, y_{P_2})$:
Coordinate x_{P_2} :

(B7CCB40627A621E2B9989403EA065CE58442FC3B14845D1A
370A8CB90980D3A6F379173E5E73249BE25AE7EDD15B39DB,
6CB21309922169AE2BD22EC4D5FC10FEB7470CDA26750225
57CDA6F9D611A0257C3E2867D0342D75C46F22BCB0856010)

Coordinate y_{P_2} :

(3F8F3F72E49333C779890EDE7B9EADC4DCCF21D516A65CAD
AAAE1209906C9D43B5E8DC93D11435A3C1C3A161A3A386D1,
F4AB6C1084256BCF6C5CFBD13393F2859F83221CA28F8F9
4004089F28C607D4B7B09172BB9625589035B90E1F0BDB13)

Example for R-ate Pairing Computation,
Computing element in G_T $g = e(P_1, P_2)$:

(7E1ACC6B5FE0ACD125BDA145891B2B2A8AAB29A307442AC1
630B2FFC2120441ABBA17DDC90EC63A901095F1F1287D9BD,
49565100D9EF20B734E8863D312F70BED296F243DA1004FB
9BF3918B55DC0088954BABDD13A9ECEB574FB3B197B81B0E,
D5D15ECF0A2ED474AA71979EB7FCF37EDE3EC9FEEE162197
AECF428BBACC708FAC790B5A2297AEE0F9463623AF578247,
1E73503EC80E80F69A439D8035D494A978DC589A4A86D969
E0E34BA0B154659A4F060A454BB5FE9E236900F467E00D3F,
D7C08D9EA24D7001C9EFD9B15B37B435328A65BC2B42C5F3
3E37176BB6492176E845226676E6C51AF461B9249248AC0D,
C0EB0A339CF12C0797FFFE43A04089CFD07B64DA9453D4B4
E4BE9EADAC5B00B69C88745CFF5C2279A4C0EE58B9F9E694,

53FDD05837CEC5FB2DEB7E07D922F37E932D44D7B3ECE754
0A131EBA0A2B6353107C39F18311EF0AACC069A97D4BCBAB,
6625857CF616CE14187D3D60B6222CE5784C2C962E166CE4
B81BED44403371ED92EDCE13772EA9595CE18DE1D20C23FF,
D2FF77AC1C4B0F4E97DE64986F80C4C19DBBBD3A2476561
773A522634E5A829260D8CF61FA6C85FF23742307710BD04,
993C7C074833ADC865D7F9240032148062E59BADB267D16A
3A6BC5B861B80608CB32EAF0F9B83908358A6983CB0A20E2,
EA2CA95D06C5DC9253842FC913F2FFD63CD7EFF5413181A6
B5283799CDCA461CD56192A13AA3D8BF2D31366490B99796,
4F6819DF53B329E0A4897EB5D11D2EC302492E7A4B55F395
14AF0C0A3CBE4B8103BF59C137999AB5AB555B1C69FF7985)

References

1. GM/T 0044.1-2016 Identity-based cryptographic algorithms SM9-Part 1: General
2. GM/T 0044.2-2016 Identity-based cryptographic algorithms SM9-Part 2: Digital signature algorithm
3. GM/T 0044.3-2016 Identity-based cryptographic algorithms SM9-Part 3: Key exchange protocol
4. GM/T 0044.4-2016 Identity-based cryptographic algorithms SM9-Part 4: Key encapsulation mechanism and public key encryption algorithm
5. GM/T 0044.5-2016 Identity-based cryptographic algorithms SM9-Part 5: Parameter definition
6. Barbulescu, R., Duquesne, S.: Updating key size estimations for pairings. J. Cryptol. **1**, 1–39 (2017)
7. Barbulescu, R., Gaudry, P., Kleinjung, T.: The tower number field sieve. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 31–55. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_2
8. Barreto, P.S.L.M., Costello, C., Misoczki, R., Naehrig, M., Pereira, G.C.C.F., Zanon, G.: Subgroup security in pairing-based cryptography. In: Lauter, K., Rodríguez-Henríquez, F. (eds.) LATINCRYPT 2015. LNCS, vol. 9230, pp. 245–265. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22174-8_14
9. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006). https://doi.org/10.1007/11693383_22
10. Barrett, P.: Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 311–323. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_24
11. Cheon, J.H.: Security analysis of the strong Diffie-Hellman problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 1–11. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_1
12. Cheon, J.H.: Discrete logarithm problems with auxiliary inputs. J. Cryptol. **23**(3), 457–476 (2009)
13. Duquesne, S., Mrabet, N.E., Haloui, S., Rondepierre, F.: Choosing and generating parameters for pairing implementation on bn curves. Appl. Algebra Eng. Commun. Comput. **1**, 1–35 (2017)

14. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. J. Cryptol. **23**(2), 224–280 (2010)
15. Fuentes-Castañeda, L., Knapp, E., Rodríguez-Henríquez, F.: Faster hashing to $\mathbb{G}_2$. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 412–430. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28496-0_25
16. Granger, R., Scott, M.: Faster squaring in the cyclotomic subgroup of sixth degree extensions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 209–223. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_13
17. Karabina, K.: Squaring in cyclotomic subgroups. Math. Comput. **82**(281), 542 (2013)
18. Kim, T., Barbulescu, R.: Extended tower number field sieve: a new complexity for the medium prime case. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 543–571. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_20
19. Knuth, D.E.: The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1997)
20. Menezes, A., Sarkar, P., Singh, S.: Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography. In: Phan, R.C.-W., Yung, M. (eds.) Mycrypt 2016. LNCS, vol. 10311, pp. 83–108. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61273-7_5
21. Montgomery, P.L.: Modular multiplication without trial division. Math. Comput. **44**(170), 519–521 (1985)
22. Montgomery, P.L.: Speeding the pollard and elliptic curve methods of factorization. Math. Comput. **48**(177), 243–264 (1987)
23. Mrabet, N.E., Joye, M.: Guide to Pairing Based Cryptography. Taylor and Francis Group, LLC (2017)
24. Naehrig, M.: Constructive and computational aspects of cryptographic pairings. Dissertation for the Doctoral Degree. Duitsland, Technische Universiteit Eindhoven (2009)
25. Pollard, J.M.: Monte Carlo methods for index computation (mod p). Math. Comput. **32**(143), 918–924 (1978)
26. Scott, M.: Implementing cryptographic pairings. In: Proceedings of the First International Conference on Pairing-Based Cryptography, Pairing 2007, pp. 177–196. Springer, Heidelberg (2007)
27. Scott, M., Benger, N., Charlemagne, M., Dominguez Perez, L.J., Kachisa, E.J.: On the final exponentiation for calculating pairings on ordinary elliptic curves. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 78–88. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03298-1_6
28. Scott, M., Guillevic, A.: A new family of pairing-friendly elliptic curves. Cryptology ePrint Archive, Report 2018/193 (2018). https://eprint.iacr.org/2018/193