Simple Method for Realizing Weil Theorem in Secure ECC Generation

Feng Hu, Chao Wang*, Huanguo Zhang, and Jie Wu

Abstract: How to quickly compute the number of points on an Elliptic Curve (EC) has been a longstanding challenge. The computational complexity of the algorithm usually employed makes it highly inefficient. Unlike the general EC, a simple method called the Weil theorem can be used to compute the order of an EC characterized by a small prime number, such as the Koblitz EC characterized by two. The fifteen secure ECs recommended by the National Institute of Standards and Technology (NIST) Digital Signature Standard contain five Koblitz ECs whose maximum base domain reaches 571 bits. Experimental results show that the computation speed decreases for base domains exceeding 600 bits. In this paper, we propose a simple method that combines the Weil theorem with Pascal's triangle, which greatly reduces the computational complexity. We have validated the performance of this method for base fields ranging from $2^{100}$ to $2^{1000}$. Furthermore, this new method can be generalized to any ECs characterized by any small prime number.

Key words: Elliptic Curves (ECs); Pascal's triangle; Weil theorem

1 Introduction

In Elliptic Curve Cryptography (ECC), the selection of parameters is critical, with the core issue being how to quickly compute the order of randomly generated Elliptic Curves (ECs)[1–7]. International scholars have researched this issue for many years. Currently, there are four main methods: the Schoof algorithm[8], SEA algorithm[9], Satoh algorithm[10], and AGM algorithm[11]. The former two are based mainly on the finite field $\mathbb{F}_q$, where $q$ is a large prime number. The other two methods are based on the finite field $\mathbb{F}_{p^m}$, where $p$ is a small prime number.

- Chao Wang and Feng Hu are with the Key Laboratory of Specialty Fiber Optics and Optical Access Networks, School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China. E-mail: wangchao@staff.shu.edu.cn; sdhf91103@163.com.
- Chao Wang and Jie Wu are with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA. E-mail: jiewu@temple.edu.
- Huanguo Zhang is with the Key Lab of Aerospace Information Security and Trusted Computing, Wuhan University, Wuhan 430072, China. E-mail: liss@whu.edu.cn.
* To whom correspondence should be addressed.

Manuscript received: 2016-11-14; accepted: 2017-02-06

Hasse theorem[12–15]: Let $k = \mathbb{F}_q, t = q + 1 - \#E(k)$, then Frobenious Endomorphism $\phi$ meets:

$$\phi \circ \phi - [t] \circ \phi + [q] = [0],$$

$$-2\sqrt{q} \leq t \leq 2\sqrt{q},$$

where $\#E(k)$ is the order of the EC.

The Hasse theorem is the foundation of the above four methods. First, we must know $t$ before we can solve for the order of the EC, using the formula $t = q + 1 - \#E(k)$. We call $t$ the curve trace. It is not easy to calculate $t$ for an EC with a large base field. The above four methods provide ways to find $t$ for different kinds of ECs[16–19].

In this paper, we focus on a special kind of EC, as shown in Table 1, which is characterized by a small prime number. Its parameters $a$ and $b$ are small numbers less than $q$. We can use a simple method called the Weil theorem to calculate their order[19–21].

Table 1 Units for magnetic properties.

| $p$ | Curve type | $(a, b)$ |
|-----|------------|----------|
| 2   | $Y^2 + XY = X^3 + aX^2 + b$ | $a \in (0, 1)$, $b \in (0, 1)$ |
| 3   | $Y^2 = X^3 + aX^2 + b$ | $a \in (0, 1, 2)$, $b \in (0, 1, 2)$ |
|     | Other small prime numbers | $Y^2 = X^3 + aX + b$, $a \in [0, p - 1]$, $b \in [0, p - 1]$ |
**Weil theorem:** The curve $E$ can be viewed as an elliptic curve over any extension field $M$ of $F_q$; $E(F_q)$ is a subgroup of $E(M)$. Let $t = p + 1 - \#E(F_p)$. Then

$$\#E(F_{q^m}) = q^m + 1 - t_e$$

(1)

$$t_e = \alpha^m + \beta^m$$

(2)

where $\alpha$ and $\beta$ are complex numbers determined from the factorization of $T^2 - tT + q = (T - \alpha)(T - \beta)$.

The implementation process of the Weil theorem, which is widely used for ECs characterized by a small prime number, can be divided into three parts, as shown in Fig. 1.

1. To calculate $t$, we must first know $\#E(F_p)$. The character base field $p$ is small, so typically, we calculate the value of $\#E(F_p)$ using the brute-force method.

2. We factorize the quadratic equation $T^2 - tT + q$ to identify its roots $\alpha$ and $\beta$. The discriminant is $D = t^2 - 4p$.

$$\alpha = \frac{t + \sqrt{t^2 - 4p}}{2}, \quad \beta = \frac{t - \sqrt{t^2 - 4p}}{2}$$

(3)

We have the relationship $t^2 < 4p$, that is, $t^2 - 4p < 0$, so $\alpha$ and $\beta$ are complex numbers that are conjugates of each other.

3. We calculate the order of $\#E(F_{p^k})$ using $k$ in Eqs. (1) and (2), where the calculation of $\alpha^m$ and $\beta^m$ in Eq. (2) is critical. Because of the complex numbers $\alpha$ and $\beta$, this calculation is too complicated without the use of a trick. In this paper, we propose a new method that combines the Weil theorem with Pascal’s triangle to reduce the computational complexity of $\alpha^m$ and $\beta^m$. This proposed method avoids the need to implement complex operations and estimate calculation accuracy.

The rest of the paper is arranged as follows. In Section 2, we introduce the basic properties of Pascal’s triangle and apply it to the Weil theorem\textsuperscript{[22,23]. In Section 3, we give corresponding algorithms for the theoretical analysis in Section 2. We present our experimental procedures, data, and analysis in Section 4. In the last section, we draw our conclusions and give the future research outlook.

## 2 Pascal’s Triangle

### 2.1 Basic properties of Pascal’s triangle

In China, Pascal’s triangle is called Yanghui triangle. Here we discuss the properties of Pascal’s triangle\textsuperscript{[9]}, the model and corresponding properties of which are shown in Fig. 2. The third property and its corresponding Eq. (4) is the most important, in that we can easily infer any number in any line by the use of this property.

$$C_i^n = C_{i-1}^{n-i} + C_i^{n-i}, \quad i = 0, 1, 2, ..., \quad n \in [1, i-1]$$

(4)

where $i$ denotes the current line, $n$ denotes the current position in the line, and $C_i^n$ denotes the number at the $n$-th position in Line $i$.

As shown in Fig. 2, Pascal’s triangle has the following properties: (1) Symmetry, the Pascal’s triangle coefficient table is symmetrical in its structure. (2) There are $n$ numbers in Line $n$ and the sum of Line $n$ is equal to $2^n - 1$. (3) Each number in Line $n$ is equal to the sum of two numbers in Line $n - 1$.

We use Pascal’s triangle to solve for the arbitrary power of the binomial expansion coefficients. There are two main forms for the binomial expansion: $(1 + x)^m$ and $(1 - x)^m$. The models of these two binomial expansions by coefficients have a similar structure to the model of Pascal’s triangle shown in Fig. 2. This similarity provides an easy way to calculate the coefficients of these two binomial expansions without the need to use multiplication.
Fig. 3 Models of the two binomial expansions.

shows the corresponding models of these two binomial expansions.

However, there is a shortcoming associated with this method. The third property is the relationship between two adjacent lines. However, we have no direct relationship between any two lines. To calculate the coefficients in Line \( n \), we must individually calculate the coefficients from Line 1 to Line \( n \). We must also compute from scratch each time.

The coefficients of these two models are constant for each computation. So, to overcome this shortcoming, we use some preprocessing tools. If a certain number of coefficients could be recorded in advance, we would then only have to query work while calculating the coefficients in any line.

2.2 Weil theorem combined with Pascal’s triangle

Our core idea is simple. If \( \alpha^m \) and \( \beta^m \) can be transformed into a form similar to \((1+x)^m\) or \((1-x)^m\), we can solve them using Pascal’s triangle and thereby greatly improve the efficiency.

Since the base field of \( E(F_p) \) is small, we can easily determine its order using the brute-force method. Then, we can calculate \( \alpha \) and \( \beta \) using Eq. (3). We show \( \alpha^m \) and \( \beta^m \) below.

Let \( d = |t^2 - 4p| \), then

\[
\alpha^m + \beta^m = \left( \frac{t}{2} \right)^m \left[ \left( 1 + \frac{\sqrt{d}}{t} \right)^m + \left( 1 - \frac{\sqrt{d}}{t} \right)^m \right]
\]  

(5)

Let \( x = \frac{\sqrt{d}}{t} \), then

\[
\alpha^m + \beta^m = \left( \frac{t}{2} \right)^m \left[ (1 + x)^m + (1 - x)^m \right]
\]  

(6)

In Eq. (6), \( \alpha^m + \beta^m \) has been transformed into a special kind of binomial expansion. Next, we design a new Pascal’s triangle model to express the special binomial expansion shown in Fig. 4. We analyzed the shortcomings of Pascal’s triangle above. To avoid double counting, we record an adapted number of coefficients to simplify the query. In addition, to save storage space, we can record only Line \( m \), which satisfies the condition that be a prime number, since we can ensure that \( \alpha^m + \beta^m \) will be real numbers only if \( m \) is a prime number.

Then, we substitute \( x \) with \( \frac{\sqrt{d}}{t} \).

\[
\alpha^m + \beta^m = \left( \frac{t}{2} \right)^m \left[ \left( 1 + \frac{\sqrt{d}}{t} \right)^m + \left( 1 - \frac{\sqrt{d}}{t} \right)^m \right]
\]  

(7)

Because the floating point calculation is too complicated to directly calculate the value, we make some adjustments to Eq. (7), as follows:

\[
\alpha^m + \beta^m = \left( \frac{t}{2} \right)^m \left[ c_{m-1} \left( \frac{\sqrt{d}}{t} \right)^{m-1} + c_{m-2} \left( \frac{\sqrt{d}}{t} \right)^{m-2} + \ldots \right]
\]  

(8)

where \( c_{m-1}, c_{m-2}, \ldots \) are the coefficients of \((1+x)^m + (1-x)^m\). We calculate these coefficients in advance so that we can find them according to the value of \( m \). Since \( c_{m-2i} (i = 0, 1, 2, \ldots) \) is equal to zero, we can further improve Eq. (8) as follows:

Fig. 4 Model of the special binomial expansion \((1+x)^m + (1-x)^m\).
\[ \alpha^m + \beta^m = \left( \frac{m}{2} \right)^m \left[ c_{m-1} \left( \frac{\sqrt{d}}{t} \right)^{m-1} + c_{m-2} \left( \frac{\sqrt{d}}{t} \right)^{m-2} \right] + \left( \frac{m}{2} \right)^m \left[ \frac{m^2}{2} \left( \frac{m-1}{2} \right) \left( \frac{m-2}{2} \right) \right] \]

Algorithm 1: Constructing model in Fig. 4c requires addition, so we have no multiplication operation in Algorithm 1. The scalar multiplication in Algorithm 1 can be regarded as an operation that realizes the function of Eqs. (8) and (1). The scalar multiplication then store the result for easy query. Algorithm 2 uses Algorithm 1 to preprocess model in Fig. 4c and use Algorithm 1 to preprocess model in Fig. 4c and uses a total of 7.6 GB of data.

We have proposed a simple method for realizing the Weil theorem. In this section, we introduce the two main algorithms used to perform this method. We use Algorithm 1 to preprocess model in Fig. 4c and then store the result for easy query. Algorithm 2 realizes the function of Eqs. (8) and (1). The scalar multiplication in Algorithm 1 can be regarded as an addition, so we have no multiplication operation in Algorithm 1. Constructing model in Fig. 4c requires that we perform addition only \(m(m-1)/2\) times and scalar multiplication \(m(m-1)/4\) times when \(m\) is even, or \(m^2 - 2m + 1/4\) times when \(m\) is odd. Compared with the direct method, the efficiency is greatly improved.

### 3 Experimental Procedure and Result Analysis

#### 3.1 Implementation of Pascal’s triangle for model in Fig. 4c

CPU: AMD sempron 2800+; memory: 256 MB; hard: 80 GB; software platform: Microsoft Visual C++ 2005.

We preprocess model (c) in Fig. 4c from \(m = 5\) to 10000, then store the coefficients of the line whose \(m\) is a prime number to save storage space, because the safe base field \(F_{p^m}\) is based on the advance condition that is a prime number. Using the above PC configuration, this process takes about twenty minutes and uses a total of 7.6 GB of data.

To construct the model (c) in Fig. 4c, we add each coefficient in each line in model (a) (Fig. 4a) to the corresponding coefficient in the corresponding line in model (b) (Fig. 4b). The new model is shown in model (c) (Fig. 4c).

In addition, the base field is considered to be safe if it meets the following two conditions.

First, the order of the EC should contain a large prime factor \(n\). We refer to \(n\) as the order of the base point. \(h\) is the cofactor, which is equal to the order of the curve divided by \(n\). The National Institute of Standards and Technology (NIST) Digital Signature Standard\(^{[24]}\) describes the size factor \(h\), which is equal to 2 or 4 when the EC is characterized by 2.

The safe base field \(F_{p^m}\) must be more than 160 bits. This requirement is used to avoid using the brute-force method, which would have no effect when the base field is sufficiently large.

In the next section, for a more in-depth analysis, we provide our experimental results for ECs characterized
by 2 and 3. For other ECs characterized by any small prime number, our method is also applicable.

### 3.2 Selection of safe base field containing a large prime factor, based on the Koblitz EC

The safe base field has the following expression:

\[
y^2 + xy = x^3 + ax + b,
\]

where \((a, b) \in (0, 1)\). When \(b = 0\), the curve trace \(t = 0\), so this curve is a super-singular EC, which is less secure, and should not be used in practice. When \(b \neq 0\), the ECs have the following expressions:

\[
y^2 + xy = x^3 + x + 1 \tag{9}
\]

\[
y^2 + xy = x^3 + 1 \tag{10}
\]

These two curves are the Koblitz\(^{[25]}\) ECs. Next, we calculate the order based on different base field \(F_{pm}\) value, where \(m\) is a prime number ranging from 1 to 10,000. Then, we divide the order of EC by \(h\) to calculate \(n\). We then check whether or not \(n\) is a large prime number. If not, we discard it. Otherwise, we obtain a safe base field.

To date, we have identified a total of 54 different general fields, which are listed in Table 2. Of course, 33 safe base fields (in **bold** type) exceed 163 bits. The safe base field \(2^{163}\) is recommended by the NIST for curve described by Eq. (9) and the safe base fields \(2^{283}, 2^{349},\) and \(2^{163}\) are recommended by the NIST for the curve described by Eq. (10). Other base fields listed in Table 2 are newly identified. We apply the same guidelines to them and have reason to believe that they are also safe. Table 3 lists the corresponding calculation times.

### Table 2 Safe base fields for Koblitz ECs.

| \(t\) | Curve-(a, b) | \(h\) | Prime number \(n\) |
|---|---|---|---|
| \(-1\) | (0,1) | 4 | 5, 7, 13, 19, 23, 41, 83, 97, 103, 107, 131, 233, 239, 277, 283, 349, 409, 571, 1249, 1913, 2221, 2647, 3169, 3527, 4349, 5333, 5903, 5923, 6701, 9127, 9829 |
| \(1\) | (1,1) | 2 | 5, 7, 11, 17, 19, 23, 101, 107, 109, 113, 163, 283, 311, 331, 347, 359, 701, 1153, 1597, 1621, 2063, 2437, 2909, 3319, 6011 |

As we can see from the above data, our results are fully consistent with the five basic point orders published by the NIST. Currently, we have completed our calculations of the basic point orders of the Koblitz EC, and the maximum order size achieved thus far is 9829, which belongs to the curve \(y^2 + xy = x^3 + 1\), shown in the Appendix.

Tables 4 and 5 show the base point orders of two Koblitz ECs for base fields ranging from \(2^{163}\) to \(2^{800}\).

### 3.3 Selection of safe general fields containing a large prime for ECs characterized by \(3\)

Unlike ECs characterized by 2, the expression for this type of EC is \(y^2 = x^3 + ax^2 + b\), where \(a, b \in (0, 2)\). When \(a = 0, b = 1\) or 2, the curve trace \(t = 0\), so this is a super-singular EC, which is less secure, and should not be used in practice. There are four non-super singular ECs for which we calculated the order in the general field \(3m\) with prime numbers \(m\) ranging from 2 to 10,000. We then checked whether or not the order of the base point was a large prime number.

To date, we have identified a total of 57 different general fields, which are listed in Table 6, of which there are 35 safe base fields (in **bold** type) that exceed 103 bits \((3^{103} \approx 2^{163})\). Table 7 lists the corresponding calculation times.

Here, we give the base point order of the EC \((a = 1, b = 2)\) on base field \(3^{991}\):

\[
22389385416870846871080886378298436315016715486327425827108339992640259965535395024103945039002556587608979444061412656454039562069110600778044086402303464201260397267722652001130827430859815989947333868289283560681818218260466890951686862789957369966227369032675323278924793093019917719662703548035599189550647956626733515794385342509610153931402126748757020050980632694631197604631172363634095994054051702005075657822740175195402269736773157042740209933430001837490296677.

### Table 3 Time consumption of \(2^m\).

| \(m\) | Time (ms) | \(m\) | Time (ms) | \(m\) | Time (ms) |
|---|---|---|---|---|---|
| 5 | 0.1 | 131 | 41 | 571 | 210 |
| 13 | 1.1 | 233 | 78 | 701 | 303 |
| 41 | 16 | 283 | 94 | 1153 | 470 |
| 83 | 28 | 349 | 110 | 1249 | 506 |
| 103 | 31 | 409 | 150 | 1597 | 570 |
| \(m\) | Time (ms) | \(m\) | Time (ms) | \(m\) | Time (ms) |
| 5 | 5903 | 6.593 | 13 | 6701 | 12.14 |
| 41 | 9127 | 28.53 | 83 | 9829 | 35.70 |
Table 4 Orders of the Koblitz EC \((a=0, b=1, \text{Eq. (10)})\).

| \(m\) | Order |
|------|-------|
| 2\(^{233}\) | 3 450 873 173 395 281 893 717 377 931 138 512 760 570 940 988 862 252 126 328 087 024 741 343 |
| 2\(^{239}\) | 220 855 883 097 298 041 197 912 187 592 864 814 948 216 561 321 709 848 887 480 219 213 21 |
| 2\(^{277}\) | 60 708 402 882 054 033 466 233 184 588 234 965 832 321 570 786 508 764 884 175 561 891 622 165 064 650 683 |
| 2\(^{283}\) | 3 885 337 784 451 458 141 838 923 813 647 037 813 284 811 733 793 061 324 295 874 997 952 819 829 704 422 603 873 |
| 2\(^{2349}\) | 286 687 326 998 758 938 951 352 611 912 760 867 599 570 623 646 034 147 884 067 443 354 153 078 850 115 990 757 740 209 812 671 |
| 2\(^{2409}\) | 330 527 984 395 124 299 479 283 677 847 627 294 075 626 569 631 244 830 993 521 422 749 282 851 602 622 232 822 777 |
| 2\(^{2776}\) | 60 708 402 882 054 033 466 233 184 588 234 965 832 321 570 786 508 764 884 175 561 891 622 165 064 650 683 |
| 2\(^{283}\) | 3 885 337 784 451 458 141 838 923 813 647 037 813 284 811 733 793 061 324 295 874 997 952 819 829 704 422 603 873 |

Table 5 Orders of the Koblitz EC \((a=1, b=1, \text{Eq. (9)})\).

| \(m\) | Order |
|------|-------|
| 2\(^{163}\) | 5 846 006 549 323 611 672 814 741 753 598 448 348 329 118 574 063 |
| 2\(^{283}\) | 7 770 675 568 902 916 283 677 847 627 294 075 626 569 631 244 830 993 521 422 749 282 851 602 622 232 822 777 |
| 2\(^{311}\) | 2 085 924 839 766 513 752 338 884 393 911 202 341 482 140 609 642 324 345 392 880 711 289 149 051 673 258 477 584 014 232 812 |
| 2\(^{331}\) | 2 187 250 724 783 011 475 767 630 354 939 786 236 916 703 635 071 711 166 739 891 218 584 916 354 726 654 294 |
| 2\(^{347}\) | 143 343 663 499 379 469 475 767 305 956 380 433 799 785 311 823 017 565 728 537 420 307 240 763 803 325 774 |
| 2\(^{359}\) | 587 135 645 693 458 306 972 374 176 217 117 621 365 353 169 430 893 227 644 473 010 306 711 358 712 586 776 588 |
| 2\(^{313}\) | 5 260 135 901 548 373 507 240 698 882 880 128 665 550 339 802 823 173 859 498 280 903 068 732 154 297 080 822 |
| 2\(^{701}\) | 113 666 536 277 588 451 226 980 007 447 205 738 750 785 915 445 464 713 273 053 067 741 405 968 564 334 794 313 753 878 032 816 084 302 756 649 401 756 057 061 240 038 011 |

Table 6 Safe base fields of ECs characterized by 3.

| \(t\) | Curve-\((a, b)\) | Prime number \(m\) |
|------|-----------------|------------------|
| 1    | 3.4 Analysis    |                  |
|      |                 |                  |

3.4 Analysis

From above data, we find that the density distribution of the safe base field of ECs decreases as \(m\) increases. Figure 5 shows the density distribution of the safe base fields of the Koblitz ECs.

As shown in Fig. 5, the colorless bars in Fig. 5 denote the number of safe fields of \(y^2 + xy = \chi^3 + x + 1\). The dark bars denote the number of safe fields of \(y^2 + xy = x^3 + 1\).

The safe base fields are mainly concentrated in the area ranging from 160–3500 bits. The density is low when the size exceeds 3500 bits.

4 Conclusion

In combination with Pascal’s triangle, we have greatly improved the efficiency of the Weil theorem by making floating-point and complex calculations unnecessary.
We have completed searches for safe fields of ECs characterized by small prime number ranging from 2 to 7. With respect to Koblitz ECs, we calculated the order up to the maximum base field $2^{10000}$. Our results cover the five safe base fields $2^{163}$, $2^{233}$, $2^{283}$, $2^{409}$, and $2^{571}$, as recommended by the NIST Digital Signature Standard. In this paper, due to space limitations, we listed our some results for ECs characterized by 2 and 3. There are few articles reporting the existence of other safe base fields whose size exceeds $2^{800}$.

Appendix

As we can see from the above data, our results are fully consistent with the five base point orders published by the NIST. Currently, we have finished the work for calculating the base point orders of the Koblitz curve ranging from 160 to 10 000. The maximum size of order is on the base field $2^{9829}$ which belongs to the curve $y^2 + xy = x^3 + 1$.

| Number of safe fields | Size of safe field (bit) |
|-----------------------|-------------------------|
| 895 487 092 948 941 362 007 327 004 453 860 902 814 |
| 302 173 251 599 495 947 261 408 799 440 521 691 212 |
| 118 367 355 336 460 768 789 429 408 810 576 343 305 |
| 302 218 988 678 212 117 788 143 145 250 255 322 |
| 460 929 527 884 596 553 270 090 098 894 953 736 784 |
| 361 399 814 150 097 980 595 188 229 450 393 729 870 |
| 645 156 496 796 663 247 745 222 921 006 399 112 326 |
| 581 852 342 622 909 179 299 952 248 469 222 401 368 |
| 860 217 278 931 113 035 926 612 546 416 955 341 142 |
| 918 360 957 636 667 128 420 482 970 699 794 463 621 |
| 178 567 078 333 813 688 109 246 954 410 348 327 913 |
| 246 004 370 881 767 919 770 418 512 471 022 605 |
| 016 531 130 206 465 862 409 385 273 321 660 586 377 |
| 953 871 508 548 640 508 914 560 833 701 824 159 360 |
| 494 384 963 823 034 715 091 292 154 987 016 874 310 |
| 190 004 158 889 582 817 203 381 770 859 465 822 288 |
| 331 582 386 284 240 223 787 390 306 145 934 450 226 |
| 489 596 924 827 049 704 008 791 765 749 111 081 724 |
| 278 493 392 746 540 472 486 811 523 842 884 983 093 |
| 844 778 513 148 529 007 957 402 464 024 736 050 213 |
| 764 098 677 174 341 055 936 821 768 666 725 283 197 |
| 113 584 722 807 257 075 492 247 106 178 870 353 847 |
| 622 481 653 284 471 096 056 827 487 389 216 315 405 |
| 105 088 864 482 904 163 335 774 821 074 009 117 425 |
| 614 295 812 594 272 641 701 795 260 007 606 643 917 |
| 789 070 194 105 317 137 469 289 553 360 477 823 953 |
| 086 311 611 349 328 104 795 207 518 135 729 277 151 |
| 378 071 385 409 900 982 806 289 928 801 751 492 106 |
| 626 261 036 072 460 270 186 769 271 183 868 484 728 |
| 004 162 580 045 528 227 555 209 990 657 659 756 224 |
| 262 618 973 685 359 687 192 857 979 912 614 472 771 |
| 047 178 649 617 836 923 871 319 131 525 241 153 623 |
| 026 452 814 149 770 261 570 616 509 235 656 657 040 |
| 282 700 236 716 011 874 005 410 920 507 986 548 748 |
| 897 370 555 842 420 446 182 114 639 999 708 229 530 |
| 574 395 112 365 803 286 047 894 361 752 069 568 852 |
| 173 409 736 432 211 741 542 074 035 675 171 302 301 |

Fig. 5 Density distribution of safe base fields of Koblitz ECs.
This research was supported by the National Natural Science Foundation of China (Nos. 61372111 and 61572304), the Innovation Grants of Shanghai Municipal Education Commission (Nos. 14ZZ089, and 61272056, and 60970006), the Innovation Grant of Shanghai Municipal Education Commission (No. 14ZZ089), and Shanghai Key Laboratory of Specialty Fiber Optics and Optical Access Networks (No. SKLSFO2014-06).

Acknowledgment

This research was supported by the National Natural Science Foundation of China (Nos. 61372111, 61572304, 61272056, and 60970006), the Innovation Grant of Shanghai Municipal Education Commission (No. 14ZZ089), and Shanghai Key Laboratory of Specialty Fiber Optics and Optical Access Networks (No. SKLSFO2014-06).

References

[1] P. Laiswal, A. Kumar, and S. Tripathi, Design of secure group key agreement protocol using elliptic curve cryptography, present at 2014 International Conference on High Performance Computing and Applications (ICHPCA), IEEE, Bhubaneswar, India, 2014.

[2] C. Wang, H. G. Zhang, and L. L. Liu, Koblitz elliptic curves generating based on evolutionary cryptography theory and verifying parameters recommended by NIST, China Communications, vol. 8, no. 4, pp. 41–49, 2011.

[3] C. Wang, H. G. Zhang, and L. L. Liu, Evolutionary cryptography theory based generating method for a secure Koblitz elliptic curve and its improvement by a hidden Markov models, Science China Information Sciences, vol. 55, no. 4, pp. 911–920, 2012.

[4] R. Guo, Q. Wen, and Z. Jin, Pairing based elliptic curve encryption scheme with hybrid problems in smart house, presented at Fourth International Conference on Intelligent Control and Information Processing, Beijing, China, 2013.

[5] J. W. Xu, K. D. Wang, C. Wang, F. Hu, Z. H. Zhang, S. Y. Xu, and J. Wu, Byzantine fault-tolerant routing for large-scale wireless sensor networks based on fast ECDSA, Tsinghua Science and Technology, vol. 20, no. 6, pp. 627–633, 2015.

[6] H. H. Jia, C. Wang, and J. Gu, Error bit correction of ECC attack based on grover quantum intermediate encounter search algorithm, Netinfo Security, no. 6, pp. 28–34, 2016.

[7] Y. H. Chen, H. H. Jia, and L. Y. Jiang, ECC scanning attack based on grover algorithm, Netinfo Security, no. 2, pp. 28–32, 2016.

[8] R. Schoof, Elliptic curves over finite fields and the computation of square roots mod $P$, Mathematics of Computation, vol. 44, no. 170, pp. 483–494, 1985.

[9] N. Elkies, Elliptic and modular curves over finite fields and related computational issues, presented at the Computational Perspectives on Number Theory, Chicago, IL, USA, 1995.

[10] T. Satoh, The canonical lift of an ordinary elliptic curve over a finite field and its point counting. Journal of the Ramanujan Mathematical Society, vol. 15, no. 4, pp. 247–270, 2000.

[11] R. Harley, Counting points with the arithmetic-geometric mean, PhD dissertation, University of Reading, Whiteknights, UK, 1996.

[12] M. Wang, G. Dai, and H. Hu, Selection of security elliptic curve based on evolution algorithm, in Proc. Int. Conf. Computational Intelligence and Natural Computing, Wuhan, China, 2009.

[13] J. Silverman, The Arithmetic of Elliptic Curves. World Book Inc, 1986.

[14] G. Shen and Y. Zhen, Application of elliptic curve cryptography in node authentication of Internet of Things, in Proc. Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China, 2013.

[15] K. Sridhar, M. Raguram, and B. Prakash, Secured elliptic curve cryptosystems for scan based VLSI architecture, in Proc. Int. Conf. Information Communication and Embedded Systems, Chennai, Thailand, 2014.

[16] K. Lui and S. Ko, Scalable elliptic curve cryptosystem FPGA processor for NIST prime curves, IEEE Transactions on Very Large Scale Integration Systems, vol. 23, no. 11, pp. 2753–2756, 2015.

[17] N. Thiranant, Y. Lee, and H. Lee, Performance comparison between RSA and elliptic curve cryptography-based QR Code Authentication, in Proc. Int. Conf. Advanced Information NETWORKING and Applications Workshops, Gwangju, South Korea, 2015.

[18] M. Indaco, F. Lauri, and A. Miele, An efficient many-core architecture for elliptic curve cryptography security assessment, in Proc. Int. Conf. Field Programmable Logic and Applications, Imperial College, London, UK, 2015, pp. 1–6.

[19] D. He and S. Zeadally, An analysis of RFID authentication schemes for Internet of Things in healthcare environment using elliptic curve cryptography, IEEE Internet of Things Journal, vol. 2, no. 1, pp. 72–83, 2015.

[20] A. Reyhanimasoleh, Parallel and high-speed computations of elliptic curve cryptography using hybrid-double multipliers, IEEE Transactions on Parallel & Distributed Systems, vol. 26, no. 6, pp. 1668–1677, 2015.
[21] P. Jilna, P. Deepthi, and S. Sameer, FPGA implementation of an elliptic curve based integrated system for encryption and authentication, in Proc. Int. Conf. Signal Processing, Informatics, Communication and Energy Systems, Kozhikode, India, 2015, pp. 1–6.

[22] S. Daniel, L. Robert, and N. Oldham, Optimization of elliptic curve operations for ECM using double & add algorithm, in Proc. Int. Conf. E-Technologies and Networks for Development, 2015, pp. 2835–2841.

[23] X. Zhang, S. Ma, and D. Han, Implementation of elliptic curve Diffie-Hellman key agreement scheme on IRIS nodes, in Proc. Int. Conf. International Conference on Intelligent Computing and Internet of Things, Harbin, China, 2015, pp. 160–163.

[24] Digital Signature Standard (DSS), Federal Information Processing Standard Publication 186-3, http://csrc.nist.gov/publications/fips/fips186-3/fips186-3.pdf, 2009.

[25] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, vol. 48, no. 177, pp. 203–309, 1987.

Jie Wu is the Chair and the Laura H. Carnell Professor at the Department of Computer and Information Sciences at Temple University. Prior to joining Temple University, USA, he was a program director at the National Science Foundation and a distinguished professor at Florida Atlantic University. He received the PhD degree from Florida Atlantic University in 1989. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Computers, IEEE Transactions on Service Computing, and Journal of Parallel and Distributed Computing. He was the general co-chair/chair for IEEE MASS 2006 and IEEE IPDPS 2008 and program co-chair for IEEE INFOCOM 2011. Currently, he is serving as a general chair for IEEE ICDCS 2013 and ACM Mobi Hoc 2014, and program chair for CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). He is a CCF Distinguished Speaker and a fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Feng Hu received the BS degree from Xi’an University of Science and Technology in 2008. Currently, he is a PhD student at Electronic and Information Engineering Dept. of Shanghai University. His research interests include information security and quantum computing cryptography.

Huanguo Zhang is currently a professor and PhD supervisor at the School of Computer of Wuhan University. He graduated from Xidian University in 1970 with a bachelor degree. His research interests include information security and trusted computing.

Chao Wang received the PhD degree from Tongji University in 1999. Currently, he is an IEEE senior member, council member of China Association of AI, Information Security Committee Vice Chair of China Electronics Institute, Committeeman of the Sixth Shanghai Expert Committee for Informatization, Directorate of China Artificial Intelligence Institute, Committeeman of CCF, IEEE Shanghai Section Secretary, IEEE Shanghai CAS Chapter Vice Chair, and IEEE Shanghai Computer Chapter Vice Chair. His research interests include wireless sensor network, network information security, and ECC quantum computing cryptography.