Efficient Electronic Payment Systems by Using a Sparse Elliptic Curve Cryptography
Essam Al-Daoud, Khalid Al-Tahat and Hamed Al-Fawareh
Pages 92-97

Efficient Electronic Payment Systems by Using a Sparse Elliptic Curve Cryptography

Essam Al-Daoud, Khalid Al-Tahat, Hamed Al-Fawareh
Faculty of Science, Computer Science Department,
Zarka Private University, Jordan.

Abstract: *This paper introduces new techniques and algorithms to reduce the size of the elliptic curve parameters. The basic idea is to use sparse coefficients and sparse base points. The sparse elements are introduced with a compact representation, thus the public key parameters are reduced about 37- 49 percent. The elliptic curve application such as e-payment and e-commerce can be implemented with better performance using the suggested approach.*

Keywords: *Elliptic Curve Cryptography, Projective Coordinate, Sparse Elements, Electronic Payment*.

Received: *March 13, 2004* | **Revised:** *January 31, 2005* | **Accepted:** *February 21, 2005*

1. Introduction

The main advantage of using the finite group of elliptic curve (EC) is that its discrete logarithm problem is believed to be harder than the discrete logarithm problem for the multiplication group of a finite field. There is no known sub-exponential algorithm that can be applied to the elliptic curve discrete logarithm problem. Another advantage that makes elliptic curves more attractive is the possibility of optimizing the arithmetic operations in the underlying field [9]. This has led to appearance of several elliptic curve cryptography products such as Security Builder, SSL Plus, WTLS Plus, TrustPoint etc. In addition many companies have purchased licenses to use EC codes and embed them in their products [14, 18, 11].

By using elliptic curve cryptosystem ECC we can use smaller key size with the same level of cryptographic security for DSA or RSA, whereby we will get smaller public key certificates, faster implementation, lower power requirements and smaller hardware processors [19, 2]. Subsequently ECC can be applied to many systems and applications [3, 1]

Elliptic curve cryptography applications and protocols rely on the elliptic curve group operations such as adding, doubling and scalar multiplication, which will not be feasible unless a suitable elliptic curve finite group and efficient underlying finite field operations are used. Thus any enhancement in the underlying finite field operations will speed up all the EC applications [4, 6]. Our approach to enhance the operations is the high utilization of the sparse elements in GF (2^n). Several new algorithms are introduced such as selecting random sparse elements algorithm, finding sparse base points, compressing and decompressing the sparse elements. This new approach does not reduce the security to the fact that the elliptic curves over GF (2^n) with sparse coefficients are isomorphic to the curves which have coefficients selected randomly. Furthermore, although the base points are restricted to be sparse; the number of sparse base points is still very huge and provides the users with rich choices. The experiment shows that the result of this improvement varies from one protocol to another based on the rate of using the base point, the number of transited bits, the key size and the ratio of doubling to adding.

The remainder of this paper is organized as follows. Section 2 presents the most efficient elliptic curves projective coordinate formulas. In Section 3 we introduce the algorithms to select and to compress the sparse elements. Moreover we discuss the abundance of the sparse points. Section 4 discusses the electronic payment models. Finally in Section 5 we show the improvement in the sparse elliptic curve electronic payment models.

2. EC Projective Coordinate Operations

In order to find the sum of two distinct points on the elliptic curve *E over (GF (2^n))* by using affine coordinates, one inverse and one multiplication are needed, but to double a point one inverse and two multiplications are required. Since the implementation of elliptic curve operation indicates that the inverse operation is still more expensive than a field multiplication, where Hankerson and others show that the cost – ratio of the inversion to the multiplication over polynomial basis is 1- 10 [12, 13, 15]. Thus, the projective coordinates *X, Y and Z* on the curve $y^2 + xy = x^3 + a_2 x^2 + a_6$ over *GF (2^n)* are used to replace the inverse operation by multiplications such that [16]:

Formula 1: $(X_1, Y_1, 1) + (X_2, Y_2, Z_2) = (X_3, Y_3, Z_3)$,

Where

$$U = Z_2^2 Y_1 + Y_2 ,$$
$$S = Z_2 X_1 + X_2 ,\ T = Z_2 S ,\ Z_3 = T^2 ,$$
$$V = Z_3 X_1 ,\ X_3 = U^2 + T(U + S^2 + Ta_2) ,$$
$$Y_3 = (V + X_3)(TU + Z_3) + Z_3^2 C .$$

Formula 1 needs 9 field multiplications and 8 temporary variables are required. López and Dahab introduce a new doubling formula which requires 5 field multiplications as follows [17, 5, 20]:

Formula 2: $2(X_1 ,Y_1 ,Z_1) =(X_2 ,Y_2 ,Z_2)$

where

$$Z_3 = Z_1^2 X_1^2 ,$$
$$X_3 = X_1^4 + a_6 Z_1^4 ,$$
$$Y_3 = a_6 Z_1^4 Z_3 + X_3(a_2 Z_3 + Y_1^2 + a_6 Z_1^4) .$$

3. Sparse Elements

This section introduces algorithms to select random curves have sparse coefficients and sparse base points. The complexity analysis for the suggested algorithms indicates that the time to generate sparse elements and base points is relatively ignored. Moreover, the reduction of the sparse element length is clarified.

3.1. Select Random Sparse Coefficients

The first step to find a suitable elliptic curve is to select random coefficients (a_2 and a_6) and the selection is repeated until a prospective curve is found. However, there is no security threat if the coefficients are restricted to be sparse in GF(q). Moreover the number of generated curves is still very huge. Algorithm 1 is suggested to generate random curves with sparse coefficients.

Algorithm 1 : Generate random sparse coefficients in GF(q)

Input : *The finite field $GF(2^n)$, s (the maximum number of ones).*

Output : *The sparse elements a_2 and a_6 in GF(q).*

```
1- $j \leftarrow 0$
2- For i =1 to rand ( s ).
        2.1 - $v \leftarrow$ rand ( n ).
        2.2 - If $x_v = 1$ Then $i \leftarrow i - 1$   ($x_v$ is the $v^{th}$ bit in the element x )
              Else $x_v = 1$.
3- if j=0 then $a_2 \leftarrow x$ , $j \leftarrow 1$ ,$x \leftarrow 0$ and goto step2
   Else $a_6 \leftarrow x$
4- Return $a_2$ and $a_6$ .
```

3.2. The Upper Bound of Sparse Base Points

This subsection shows that even if the base point is restricted to be sparse, the number of generated points is still very huge.

Definition. Let *G* be a point on *E(GF (2^n))* represented in the binary expand. Then the point *G* is sparse if and only if the first coordinate has a few ones such that the number of ones is less than 5 percent from the field size. Moreover the point *G* is called sparse with s ones if the maximum number of the ones in the first coordinate is *s*.

Theorem. Let *E* be any elliptic curve over *GF (2^n)*; then the upper bound of the sparse points with s ones on *E* is

$$2\sum_{t=1}^{s}\left(\prod_{i=0}^{t-1}(n-i)\right)/t!$$

Proof:
Let *x* be any element in GF (q) and has *t* ones, then *x* can be represented in

$$\binom{n}{t} = \frac{n!}{t!(n-t)!}$$
$$= \left(\prod_{i=0}^{t-1}(n-i)\right)/t!$$

different ways. If the maximum number of the ones in the *x* coordinate is *s*, then *x* can be represented in

$$\sum_{t=1}^{s}\left(\prod_{i=0}^{t-1}(n-i)\right)/t!$$

different ways. Since the quadratic equation has two solutions when *x* is in *GF(q)*, then the upper bound for the number of sparse points with *s* ones on *E* is

$$2\sum_{t=1}^{s}\left(\prod_{i=0}^{t-1}(n-i)\right)/t! . \blacksquare$$

The order of the selected elliptic curve must be prime or nearly prime (the curve E_j has a nearly prime order if $\#E_j = r_j\ p_j$ for small integer r_j and large prime number p_j , where *j* is an integer), then the approximately average number (if the test is run over many curves E_j) of sparse base points with *s* ones is :

$$\sum_{t=1}^{s}\left(\prod_{i=0}^{t-1}(n-i)\right)/d(t!) .$$

where d is the average of r_j . Thus, this number is large enough to give the users rich choices, for example if n =160 and s = 7 then the number of sparse base points are nearly 10^{17}. However there is no security threat known in case if many users choose the same base point.

3.3. Selecting a Sparse Base Point

Algorithm 2 is suggested to find a random sparse base point *P*, with s ones for any nearly prime elliptic curve over *GF* (2^n), where P has a large prime order.

Algorithm 2: Choosing Random sparse base point with s ones.
Input: *an elliptic curve E over GF (q), the curve order rk, and the maximum number of ones s.*
Output: *a sparse base point with s ones.*

```
1- For i =1 to rand (s ).
2- v ← rand (n ).
3- If $x_v$ = 1 Then i ← i -1   ($x_v$ is the $v^{th}$ bit
   in the element x )
       Else $x_v$ = 1.
4- End For
5- Find the coordinate y, if y does not exist
   go to step 1 else set y to one solution.
6- G← (x ,y)
7- P ← k G
8- If P = O then go to Step 1.
9- Q ← r G
10- If Q ≠ O then output "wrong order"
    and stop
11- Output G.
```

The complexity of the previous algorithm is equal to the complexity of standard algorithm to generate random base points.

3.4. Compact Sparse Elements Representation

To utilize the sparse field elements in the real communication and implementation; Algorithms 3 and 4 are introduced to compress and decompress any sparse element with *s* ones in relatively ignored time.

Algorithm 3: *Compression of any sparse element in GF(2^n), where $n \le 256$*
Input : *Sparse element x*
Output : *Compressed representation array C*

```
1- m ← 1
2- For i =1 to n
     2.1 - If $x_i \ne 1$ then continue
     2.2 - $C_m$ = i
     2.3 - m ← m + 1
3- Return C.
```

Algorithm 4: *Decompression of compact representation*
Input : *Compressed representation array C with length s.*
Output : *Sparse element x*

```
1- For i =1 to s
     1.1- t ← $C_m$
     1.2- $x_t$ ← 1
2- Return x
```

Since the discrete problem for elliptic curves with field size less than 256 is sufficient for the current applications and at least for next few years; discussion will be restricted to this field size, but it can be extended easily to any other field size. Thus each element in the array C (which forms the locations of non zero bits in a sparse element) can be represented in 8 bits, so the size of the array C is (*8s*). Table 1 shows the reduction rate of the sparse elements.

Table 1. The reduction rate of the sparse elements

Field Size (n)	s	Conventional	Compact	reduction Rate
131	6	131	48	63.4
163	7	163	56	65.7
191	5	191	40	79.1
239	5	239	40	83.3
256	5	256	40	84.3

To generate the elliptic curve public keys and private keys we have to use the following steps:

1- Select a suitable elliptic curve *E* defined over GF(q).

2- Select a base point $P \in E$(GF(q)) of order l , where l is the elliptic curve *order.*

3- Select an integer *t* in the interval [1, l - 1].

4- Compute the point $Q = t P$.

Thus, the *EC* public key is (a_2 , a_6 , *P*, l, *Q*) and *the* private key is *t* (in the EC public key we need another two bits to reconstruct EC y's coordinate from x coordinate).. If a_2, a_6 and the first coordinate of the base point P are sparse, then by using the suggested approach and algorithms; the number of pubic key bits will be reduced from *(5n +2)* for the standard setting to *(24s + 2n + 2)*. Table 2 shows the percentage of the bits reduction in the EC public key.

Table 2. The reduction rate of the bits by using the new approach

Field size n	s	PK Standard	PK Compact	Reduction rate
131	6	657	408	37.8
163	7	817	496	39.2
191	5	957	504	47.3
211	5	1057	544	48.5
239	6	1197	624	47.8
256	6	1282	652	49.1

4. Electronic Payment Models

Electronic transaction uses the cryptography for many purposes such as protect the transactions against attack on the network, ensure the security without prior arrangements between customers and vendors, guarantee the transaction integrity, authenticate the customers, and authenticate the vendors. The developer of electronic payment model assumes that the banks have full Internet connection to provide general banking services, opening account, issuing checks, insurance etc. Currently few banks support general banking services as SFNB in US (http://www.sfnb.com), and BankNet in UK. Figure (1) shows the banks with full Internet connection [22].

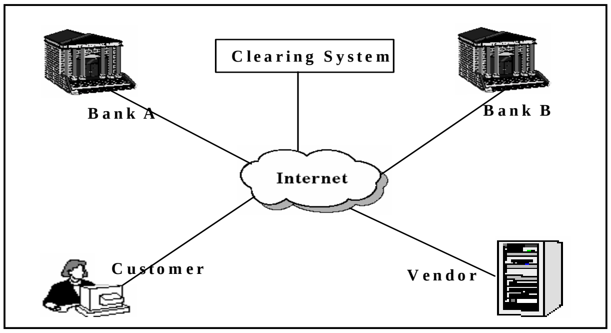


Figure 1. Clearing system with full Internet connection.

4.1. Off –Line Electronic Payment Model

In this model the payees accumulate the digital money, and then deposit it in his account when the network traffic is low. The essential components of this model are [10]:

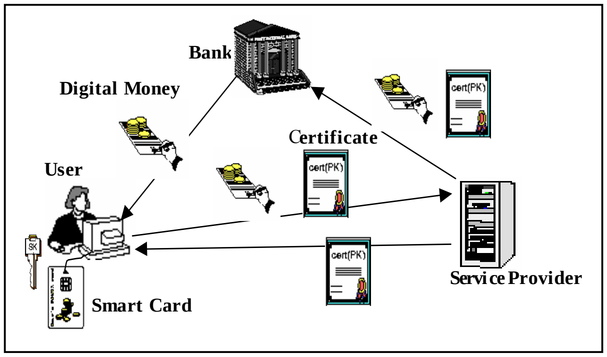


Figure 2. An off- line electronic payment model.

a) Public Key Certificates: The certificates are used to prove the relation between the user name and his public key. In figure (2) there are two certificates. The first certificate to prove the identity of the service provider (or in general the payee), it must be signed by a trusted certificate authority. The second certificate is to prove that the bank has issued the user public key. Therefore the bank must sign this certificate. The minimum contents of the certificates are the issuer name, subject name, the validity, the public key and the certificate authority signature.

b) Smart card: Smart card or tamper- resistant has been issued to the user by the bank. Three purposes of the tamper- resistant device: the first is to keep track of the user balance, hence the balance counter will be increased if the user withdraw electronically form the bank, and it will be decreased if the user pay to the service provider. The second is to validate the bank signature on amount of the money, before the balance counter is updated. The third is to sign any amount of money which will be paid to a payee. The secret keys are known just to the bank and the public keys are incorporated into the card.

c) Digital Money: The minimum contents of the digital money are the amount of the money, the identity of the payee, the serial number and the signature on these information. Additional techniques must be used to prevent the double spend and to ensure the privacy of the payment, these techniques as the serial number and blind protocol.

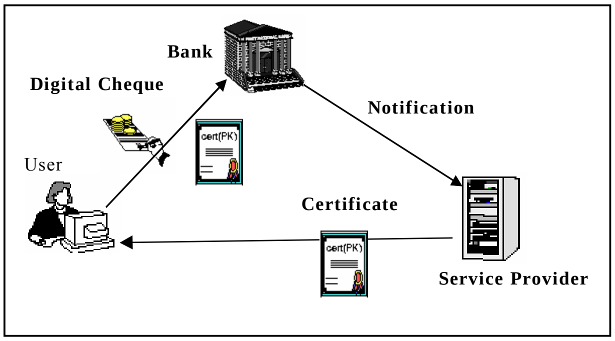


Figure 3. An on - line electronic payment model

4.2. On –Line Electronic Payment Model

Several On –Line electronic payment models have been suggested [21]. The essential components for these models are:

a) Digital Cheque: The contents of the digital Cheque like the digital money. The main different is that the amount of money will be reduced from the issuer account.

b) Public Key Certificates: Two certificates are used in Figure (3); the first is to recognize the service provider and the second is to identify the user entities.

c) Secure Socket Layer (SSL): SSL can be incorporated into this mode to exchange the data easily. SSL was originally developed by Netscape to secure the Internet transactions. Three protocols are used in SSL: the record protocol, the handshake protocol and the alert protocol [7].

5. Electronic Payments Models Analysis

In order to measure the efficiency of the electronic payment models; we will discuss four factors. The factors are: The Database Storage, the transited data, the number of verifying the information and the signatures. Table (3) and (4) summarizes these factors.

Table 3. An off-line electronic payment model analysis.

	The Bank	**The User**	**The Service Provider**
The Additional Database Storage	SK and PK (for each user)	None	Digital Money and a Certificate (for each customer).
The Sent Data	Digital Money	Digital Money and a Certificate	Digital Money and a Certificate
The Received Data	Digital Money and a Certificate	Digital Money and two Certificates	Digital Money and a Certificate
The Signatures	1 (low frequence)	1	None
The Verifying	1	2	2

Table 4. An on -line electronic payment model analysis

	The Bank	**The User**	**The Service Provider**
The Additional Database Storage	None	None	None
The Sent Data	Notification	Digital Cheque and a Certificate	A Certificate
The Received Data	Digital Cheque and a Certificate	A Certificates	Notification
The Signatures	None	1	None
The Verifying	2 (high frequence)	1	None

To compare the size of the model's objects consider the following:

1- The EC field size is 256, which equivalent to 1620 RSA key size [8].
2- The issuer name, subject name, the validity, the amount of the money, the identity of the payee and the serial number can be represented in 80 bits for each.
3- This model uses the minimum requirement contents for the digital money and the certificates.
4- The digital signature is part from the digital money, digital cheque and the certificates

Table 5. Comparison of the object size by different approaches

	RSA	**EC**	**The new approach, s = 6**
Digital Money	80*3+1620 =**1820**	80*3+256= **496**	80*3+256= **496**
Certificate	80*3+1652+ 1620 =**3472**	80*3+1280+2 56= **1776**	80*3+658+256= **1154**
Secret Key	**1620**	**256**	**256**
Public Key	**1652**	**1280**	**658**

From Table (5) we deduce that: the new approach is the best for minimizing the objects size, which will lead to enhance the performance for all parties. For example in the off-line model; the total sent data from the service provide is reduced to 31.1 percent of the data sent by using RSA and 72.6 percent of the data sent by using standard Elliptic curve. In the on-line model the total sent data from the service provide are reduced to 33.2 percent of the data sent by using RSA and 64.9 percent of the data sent by using the standard Elliptic curve. In the same way; the redaction can be shown for other factors and all parties.

By using the new approach the verifying and the signing time will be reduced for all parties. The off-line model is better for bank performance, while the on-line model is better for the service provider performance.

6. Conclusion

To satisfy efficient computations and communications, several algorithms are introduced such as selecting random sparse elements algorithm, finding sparse base points, compressing and decompressing the sparse elements. The new approaches and algorithms lead to reduce in the public key parameters by 37-49 percent and did not sacrifice in the elliptic curve cryptography security. Therefore, the elliptic curve application such as e-payment and e-commerce can be implemented with better performance using the suggested approach.

7. References

[1] Al-Daoud, E. and Ramlan, M., "Elliptic Curve Arithmetic Operations Over GF(2^n) and GF(P) For Cryptosystems Purposes," *International Conference on Mathematics and its Applications in the New Millennium*, pp381-388, 2000.

[2] Al-Daoud, E., and Ramlan, M., " A New Addition Formula For Elliptic Curves Over GF(2^n)," *IEEE Transactions on Computers.* Volume 51, Number 8, August 2002, pp 972-975, 2001.

[3] Bellare, M., et al., "Variety Cash*:* a Multi-Purpose Electronic Payment System," In *Proc. 3rd Usenix Workshop on Electronic Commerce*, Boston, 1998.

[4] Blake, I. F., Seroussi G. and N. P. Smart., *Elliptic Curve in Cryptography*, University Press, London, Cambridge, 1999.

[5] Brands, S., "Electronic Cash on the Internet," *Proceedings of the Internet Society 1995 Symposium on Network and Distributed System Security, San Diego, California,* February, 16-17, 1995.

[6] Certicom, 2003. http://www.certicom.com. Accessed on 5 march 2003.

[7] Gupta, V. S. et al., "Performance analysis of elliptic curve cryptography for Ssl," *In ACM Workshop on Wireless Security,* Atlanta, Georgia. 2002.

[8] Gura, N. et al., "Generic implementations of elliptic curve cryptography using partial reduction," *In 9th ACM Conference on Computers and Communications Security,* Washington, DC,2002.

[9] Gura, N. H. et al., "An end-to-end systems approach to elliptic curve cryptography," *In CHES '2002 Workshop on Cryptographic Hardware and Embedded Systems, Lecture Note in Computer Science. Springer-Verlag,* Redwood City, California. 2002.

[10] Ha., J. S. et al., "Compact implementation of Elliptic Curve Cryptography System using a FPGA," *The 9 th Korean conference on Semiconductors*, Feb.,21-22, pp 813- 814. 2002.

[11] Hankerson, D., et al., "Software Implementation of Elliptic Curve Cryptography over Binary Fields," *CHES' 2000 LNCS No. 1965*, pag 1-24. 2000.

[12] IEEE P1363 Draft, Standard Specifications for Public Key Cryptography, 1999. http://grouper.ieee.org/groups/1363/

[13] King, B., "An Improved Implementation of Elliptic Curves over GF(2) when Using Projective Point Arithmetic," *Selected Areas in Cryptography,* 134-150, 2001.

[14] López, J. and Dahab, R., "High-Speed Software Multiplication in GF(2^m)," *IC Technical Reports, IC-00-09, Institute of Computing, University of Campinas,* 2000.

[15] López, J. and Dahab, R., "Improved Algorithms for Elliptic Curve Arithmetic in GF(2^n)," *Selected Areas in Cryptography, 5th Annual Int. Workshop*, pp 201-212. 1998.

[16] Paar, C., "Implementation options for finite field arithmetic for elliptic curve cryptosystems," *Invited presentation at the 3rd Workshop on Elliptic Curve Cryptography (ECC '99).* University of Waterloo, Waterloo, Ontario, Canada, pp 1-3. 1999.

[17] Pedersen, T. P., "Electronic Payments of Small Amounts," *Security Protocols Workshop 59-68*, 1996.

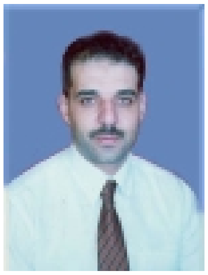
[18] Robert D. Silverman , A., "Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths," *RSA Laboratories,* 2000. http://RSA.com.

[19] Rosing, M., *Implementing Elliptic Curve Cryptography,* Manning, Greenwich. 1999.

[20] Smith, R. E., *Internet Cryptography*, Addison – Wesley, Harlow, England. 1999.

[21] Sklavos, N. and Koufopavlou, O., "Mobile Communications World: Security Implementations Aspects - A State of the Art," *Computer Science Journal of Moldova, Institute of Mathematics and Computer Science,* Vol. 31, Number 2. 2003.

[22] Weimerskirch, A., et al., "Elliptic Curve Cryptography on a Palm OS Device," *The 6th Australasian Conference on Information Security and Privacy (ACISP 2001), LNCS 2119*, Macquarie University, Sydney, Australia, pp 502-514. 2001.



Dr. Essam Al Daoud is an assistant professor at the Department of Computer Science, Zarka Private University. He received his Ph.D from University Putra Malaysia, 2001. His research interested cryptography, data mining, quantum computing, neural networks and singular value decomposition (SVD).

Dr. Khalid Al-Tahat is an assistant professor at the Department of Software engineering, Hashimate University Jordan.

Dr. Hamed J. Al-Fawareh, received his B.Sc. in Computer Science from Yarmouk University, M.Sc. in Computer Science from University Putra Malaysia, 1998 and Ph.D. in Software Engineering from University Putra Malaysia 2001. Currently he is a chairman of Computer Science Department, Zarka Private University (Jordan). research interest includes software engineering and security, and Bioinformatics. He has published many original contributions in the field of software engineering.