A Dynamic Scalable Cryptosystem Based-on Reduced Key Size ECC

Jia Wang, Lee-Ming Cheng
Department of Electronic Engineering
City University of Hong Kong
Kowloon, Hong Kong
E-mail: jwang244-c@my.cityu.edu.hk

***Abstract*—This paper proposed a dynamic scalable cryptosystem based on reduced key size Elliptic Curve Cryptography (ECC) used for low-cost mobile devices or Radio Frequency Identification (RFID)-like devices, which are extremely constrained in memory, computing power and battery life. Supplementary algorithms including dynamic parameter assigning, random base point selection and session base point synchronization are designed in order to enhance the security level of the system. Synchronization is also performed for the consistency of the curve used between the server and the client. The proposed approach will provide a composite ρ-*sec* of (110+22) = 132 >128 and requires less storage and computing power, which provides good security alternative for the wireless transformation between devices with simpler and smaller processors.**

Keywords-ECC; scalable; cryptosystem; RFID security.

I. INTRODUCTION

Recent years have seen the great convenience brought to all walks of life by various kinds of applications built around the overly general concept called the Internet of Tings (IoT). A growing number of distributed diverse embedded devices get connected to the IoT platforms, sharing the facility as well as the security threatening due to the availability of tracking devices on communicating via RF. Security assurance, e.g., access control and eavesdropping prevention, must be guaranteed on the IoT before sending out of ID; mechanisms should be established to address the privacy concerns [1][2] on IoT. Deploying of a IoT in new context [3] which is not specifically designed may also imposing threats to system.

Traditional cryptographic algorithms used in protocols proposed for transformation security, such as the widely adopted Diffie-Hellman (DH) or Rivest, Shamir and Adleman (RSA) algorithms, often involve intensive computations and create a big challenge for the embedded devices in the IoTs which are usually designed with small and simple processors for low-cost purpose. Recently, the breakthrough in analysis discrete logarithm (DL) problem by Barbulescu et al [4], Adj et al [5] and Granger et al [6] has leaded into the belief that DL based algorithms like DH and RSA will soon to break and will be phased out much earlier than its' anticipation. The new challenge facing DH and RSA over time in term of being crypto-analysis and the proliferation of smaller and simpler devices hold them back to be used in the area of IoT cryptography.

Meanwhile, Elliptic Curve Cryptographic (ECC) algorithms, which developed by Neil Kobliz and Victor Miller in 1985, requires smaller key size and less power consumption while providing equivalent security compared with other crypto-algorithms, illustrated in Table I. This feature leads to significant performance advantages especially for IoT platforms where computing power, memory and battery life of devices are extremely constrained.

However, ECC belongs to the class of stream cipher and their hardly scalable and complex in operations are the major obstacles of making them to be popular when compared with RSA. In addition, the traditional way of implementing secure elliptic curves will not be suitable to provide a solution for unified platform for a diversified devices and servers with processors ranging from 8 to 256 bits as it requires at least 200 bits key size capable scalable to 521 bits or more. In order to address the problems mentioned above, in this paper we proposed a dynamic scalable cryptosystem based on reduced key size elliptic curves with supplementary algorithms to enhance its security. The supplementary algorithms include Dynamic Parameter Assigning (DPA), Random Initial Base Point Selection (RIBPS) and Session Base Point Synchronization (SBPS). A synchronization protocol is also designed for the consistency of the curve used between the server and the client.

TABLE I. SECURITY COMPARISON OF DIFFERENT CRYPTO-ALGORITHMS

Security Equivalent	**Bit/modular bit size**					
Symmetric Algorithm	56	80	112	128	192	256
RSA&DH	512	1024	2048	3072	7680	15360
ECC	112	160	224	256	384	521

The rest of the paper is organized as follows. Section II gives an overview of ECC. The architecture of the proposed dynamic scalable cryptosystem based on reduced key size ECC is given in Section III. In Section IV we introduce the synchronization protocol designed for the consistency of the

 ISBN: 978-1-61208-396-4

transformation between the server and the clients. Simulation results are presented in Section V. The security and complexity are analyzed in Section VI. Finally we summarize the paper and discuss our future work in Section VII.

II. OVERVIEW OF ECC

In this section, we give a brief introduction of ECC. Good reference for ECC could be found in [7].

The notations used are defined as follows.

- q is the order of the underlying finite field.
- F_q is the underlying finite field of order q.
- E is an elliptic curve defined over F_q.
- $E(F_q)$ is the set of points on E both of whose coordinates are in F_q, together with the point at infinity.
- P is a point in $E(F_q)$.
- Q is another point in $E(F_q)$.
- a and b are elements of finite field F_q, c is a integer.
- n is the order of the point P.
- k is a random integer selected in the interval [2,n-2].

Elliptic curves used in cryptography are plane curves defined over finite field which consists of points satisfying the equation $y^2 = x^3 + ax^2 + \mathrm{b}$, along with the infinity point O. $E(F_q)$ together with the group operation of elliptic curves construct an Abelian group. Rules of elliptic curve arithmetic such as point addition, multiplication could be found in [7]. The order of point P is defined as the smallest positive integer n such that $nP = O$, which are usually large prime numbers in practical applications. ECC uses points in $E(F_q)$ as the elements for encryption. When a specific curve is chosen, P is randomly selected from all the points on the curve to generate the public key Q by field multiplication $Q = kP$, where k is called the private key. Q is used for encryption/signature verification and k is used for decryption/signature generation. The curve $E(F_q)$ and the base point Q are public in typical ECCs.

An example is given as simple application of ECC. Suppose *Alice* wants to sends *Bob* messages secretly. For *Alice*, she will randomly generate an integer k_a and compute $Q_a = k_aQ$ and then make Q_a public. *Bob* will randomly generate an integer k_B and compute $Q_b = k_bQ$ and make Q_b public.

To send message M_a to *Bob*, *Alice* will do:

1) Calculate the cipher $C_a = M_a + k_aQ_b$.
2) Send C_a to *Bob*.

To decrypt massage M_A from *Alice*, *Bob* will do:

1) Calculate $D_a = k_bQ_a$.
2) Decrypt $M_a = C_a – D_a$.

The security of ECC relies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), that is given E defined in F_q and two points $P, Q \in E(F_q)$, find a integer k such that field multiplication $kP = Q$, provided such an integer exists. This problem is considered computationally infeasible to solve.

Not all elliptic curves are secure. Super singular curves could be cracked much faster over none singular one and a great deal of effort have been expended at finding curves that suitable for cryptography. Aranha et al [8] proposed the use of Curve22519 and Curve1174 which are Montgomery curve and Edward curve in the form of $y^2 = x^3 + ax^2 + x$ and $x^2 + y^2 = 1 + ax^2y^2$ respectively with prime $2^b - c$.

According to [8], more secure curves can be designed using different '*a*'s and '*b*'s. A set of known secured Montgomery curves with *(a, b, c)* are given in Table II. These parameters can be utilized to building a scalable structure by presetting the parameters *(a, b, c)*, in term of different key sizes.

TABLE II. SCALABLE ECC USING VARIOUS CURVE PARAMETERS

	ρ-sec	*a*	*b*	*c*	No. (*l*)
Montgomery Curve	110	117050	221	3	0
	128	486662	255	19	1
		61370	256	189	2
		240222	256	765	3
		55790	254	$-127x2^{240}-1$	4
	192	2065150	383	187	5
	256	530438	511	187	6

TABLE III. ARCHITECTURE OF THE CURVE LIST

No. (*l*)	Curve Parameters				Initial Point (*P*)
0	a_0	b_0	c_0	n_0	P_0
1	a_1	b_1	c_1	n_1	P_1
2	a_2	b_2	c_2	n_2	P_2
……	…	…	…		……
L-1	a_{L-1}	b_{L-1}	c_{L-1}	n_{L-1}	P_{L-1}

III. ARCHITECTURE OF THE PROPOSED DYNAMIC SCALABLE CRYPTOSYSTEM

The dynamic parameter assigning is dedicated to selecting a preset group of shared parameters '*a*'s, '*b*'s and '*n*'s, which constructed a reduced secure curve, details are demonstrated in part *A*. Since a new set of curves requires computing a group of new base points, the traditional way of point generation will not be appropriate. A random initial base point selection approach [9] needs to be established. A cache will be designed to house these initial points. The Parameter Synchronization method similar to [10] is needed to provide a session base point selection.

A. Dynamic Parameter Assigning

This approach is very similar to the minimal list approach for use in RFID [11]. RFID tags store a list of random parameters or pseudonyms for authorization [11]. Each time the tag is requested, it will send out the next pseudonym in the list, going back to the beginning when the pseudonym is exhausted. The proposed DPA works similar to pseudonym flow where the DPA list holds a set of parameters and each time it is requested, the next set of parameters will be sent out and cycling back to its beginning when it hits the end of the list.

B. Random Initial Base Point Selection

Input dynamic parameters *a*, *b*, *c* and *n* (*n*>0) as integers, the algorithm will select an effective random point *P* on the desired curve and form the base point by calculating *kP*. The scalar multiplication value is used to verify the correctness of base point selection on an elliptic curve.

The base point choice algorithm of ECC on Montgomery Curve is given as an Example below.
Input: *a*, *b*, *c*, *n*, *k*.
Output: Effective base point *G*.
Steps:

1) Randomly choose *x* (0<*x*<*n*).
2) Calculate $y^2 = (x^3 + ax^2 + x) \bmod (2^b - c)$.
3) Check if v belongs to quadratic residue of $\bmod(2^b - c)$, if so y is found, select $P = (x, y)$ go to 4), if not, go to 1).
4) Compute $G = kP$, then check whether *G* meets $y^2 = (x^3 + ax^2 + x) \bmod (2^b - c)$ and *G* is not infinite point. If so, *G* is set to be the base point, and go 5), if not, go to 1) .
5) Return *G*.

As shown in Table III, each curve in the list has been allocated some space to store the initial point *P* and the value of *P* should be overwritten as *G* each time after the curve having been used for security enhancement.

C. Session Base Point Synchronization

De-synchronization will raise concern in real applications although it can provide protection on denial-of-service attacks. One possible approach to solve de-synchronization problem is to maintain a list not just of current *parameters*, but also of values from several future time-steps. This approach is similar to that of [10] involving tag resynchronization by checking plus and minus one step parameters when de-synchronization occurs. The advantage of our proposal is that it would permit a certain amount of synchronization, but would still not leak any sequence values.

IV. SYNCHRONIZATION MECHANISM OF THE SERVER AND THE CLIENTS

A. Symbol notations

- *S* server
- *C* client
- *m* Pseudo-Random Number Generated by the server's PRNG
- *r* Pseudo-Random Number Generated by the client's PRNG
- *f(x)* hash function of x
- *L* total number of curves in the list
- *l* the randomly selected curve, 0<=*l*<=*L-1*

B. Protocol Flow

As shown in Figure 1, the protocol sequences are as follows:

1) *S* generates a random number *m* of 16 bits by its PRNG and sends it to the client *C*.
2) *C* generates a random number *r* of 16 bits and sends it to *S*.
3) *S* forms the message $M_1 = r \ Xor \ m$, $M_2 = (M_1 \ mod \ L) \ Xor \ l$, and $f(M_1)$, then sends M_2 and $f(M_1)$ to *C*.
4) *C* computes $f_1 = f(r \ Xor \ m)$ and compares it with $f(M_1)$ if $f_1 = f(M_1)$ then computes $l = ((r \ Xor \ m) \ mod \ L) \ Xor \ M_2$ as the selected curve.

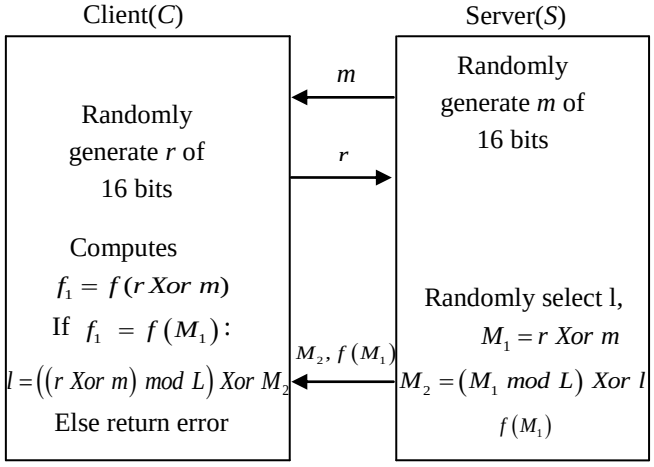

Figure 1. The proposed synchronization protocol.

An example is given here.

Suppose that there are 32 curves in the list which are numbered from 0 to 31, let $l = 11$. Assuming that $m = 0111010101010101$, $r = 1010101111001011$

First of all, the server sends *m* to the client and the client returns r back to the server. The server then computes:

M1 = r Xor m

= 1010101111001011 *Xor*
0111010101010101
= 1101111010011110
M2 = 11110 *Xor* 01011
= 10101

Suppose the transformation of *r* and *m* is performed in secure channel and f_1 equals to $f(M_1)$, then the client calculates:

(r Xor m) mod L= 11110,
$l = ((r \text{ Xor } m) \bmod L) \text{ Xor } M_2$
= 11110 *Xor* 10101
= 01011

The result indicates that the curve numbering 11 is chosen to be used by the server in the cryptography.

V. SIMMULATION RESULTS

The proposed scheme implemented on MIRACL crypto library with GCC in C programming language on Linux platforms with 2.8GHz Intel processor i7 and memory 4.0GB.

```
Start to generate the genration point.....
Genaretion of the generation point DONE!
The randomly generated point on the selected curve is:
x = 6D77D6EE2AC3E2032159FA69342DC994F696186D71C57ADAE534F83
y = 14EA1DD635781E79BE5BF06DD18110D400C6E65F241AAF0E22BAF7AE

The randomly selected value of k is:20

The generated base point G is:
x = 700A4519936C0AB5F85851F23E71E1C300541CF5FBA0FC6355B9A37
y = 330A09E9CC17F3E1A98331181035273DAE1E07B41C28EAD5C61B218

Private key of Alice ka is:19

The generated public key Q is:
x = AE448A015E0E704C31F5DA61E5AE4A62F65E3E4A78CD4A87CA34A62
y = 76B6743AF8B890259CA0838785D938A92472BEC06C10B383328B9DD

Private key of Bob kb is:9

The generated public key R is:
x = A143AA6BCFD7AAD356C76FA5D84D7628F194419D383439290A5B813
y = 36C2B738A6FBE42CBBD045FDBC754167BC2AC4814953BB9E56B25FA

The message sent by Bob is:
x = A143AA6BCFD7AAD356C76FA5D84D7628F194419D383439290A5B813
y = 36C2B738A6FBE42CBBD045FDBC754167BC2AC4814953BB9E56B25FA

The massage sent to Alice is:
x = 1D5A624E54AE15F5EFC4CC7051E5912C6A74DAE5654567186A9E67D6
y = 11C237449776620F9DD3E813AAB8A0B55D469073AD7F665B80DBA98A

The massage decrypted by Alice is:
x = A143AA6BCFD7AAD356C76FA5D84D7628F194419D383439290A5B813
y = 36C2B738A6FBE42CBBD045FDBC754167BC2AC4814953BB9E56B25FA
```

Figure 2. Simulation results of the proposed scheme

In the implementation, we generated a curve table using a set of nonsingular ellicptic curves with shorter key-sizes and then labelled them with curve numbers. By using schronizaiton mechanics proposed in Section IV, *Alice* chose a curve and shared with *Bob*. In this experiment case, the curve with *No. 0* is selected. Without loss of generality, the private keys of *Alice* and *Bob* are randomly generated as small integers for simplicity. The simulation result is shown in Figure 2. Details of time consumption of this case are inllustrated in Figure 3.

```
Time used for the generation of the Generation Point is:0.613 ms.
Time used for the generation of the Base Point is:0.497 ms.
Time used for the generation of Alice's Publilc Key is:0.441 ms.
Time used for the generation of Bob's Public Key is:0.393 ms.
Time used for the Encryption Process:0.031 ms.
Time used for the Decrypt Process is:0.184 ms.
Total time used for this case is:2.633 ms.
```

Figure 3. Time consumption of the experiment case

VI. SECURITY AND COMPLEXITY ANALYSIS

In this section, we studied the proposed scheme according to its security level, storage requirement and power consumption, which are demonstrated in part *A, B, C* respectively.

A. *Security level*

Given the security of ECC denoted by ρ-sec in term of bit size [8], from Table II it is *c*lear that in order to achieve *ρ-sec* of 128, the ECC module bit size should be around 256. By randomly selecting lower ρ-sec curves generated with different *(a, b, c)* of the same *ρ-sec*, the same high level can be achieved because the random prime choice can bridge the extra security required. Table II gives examples of various configurations of Montgomery Curves with *ρ-sec*=128 [12].

In our simulation, *ρ-sec* curves of 110 is used, the prime bit size will be 220. Table III shows the bit size required for symmetric and asymmetric crypto-algorithms. According to RSA/DH security, a total of 2^{22} primes can be found. The randomly selected prime will provide addition *ρ-sec* of 22. This approach will provide a composite *ρ-sec* of (110+22) = 132 >128.

B. *Storage requirements*

As mentioned earlier, due to the use of randomly selected curve, the traditional way to generate the base point is not feasible as all the points would be stored. We store one initial point instead of storing all the points on the curve by using RIBPS, which extremely reduces the required memory space.

C. *Efficiency and power consumption*

Reduced key size elliptic curves are used in the proposed dynamic scalable cryptosystem in order to provide flexible security mechanism for the diverse devices and servers with processors ranging from 8 to 256 bits in the IoT platforms. As illustrated in Table IV, our scheme used reduced key-size curves to gain even higher security. Shorter key-size leads to great reduction of the computation load as well as running time.

 ISBN: 978-1-61208-396-4

TABLE IV. EFFICIENCY COMPARISON

	ρ-sec	*Key-size (bit)*
Our scheme	> 132	221
Typical ECCs	128	256

VII. CONLUSION AND FUTURE WORK

A dynamic scalable cryptosystem based on reduced key size elliptic curves is proposed for the IoT platforms. Less storage and lower power consumption but higher level of security can be achieved with supplementary algorithms include dynamic parameter assigning, random initial base point selection and session base point synchronization. It could be used to address the security problems targeting IoTs, Cloud platforms and Cyber-Physical Systems, which usually connected with highly distributed low-cost smart devices, based point of view of ECC. It's expected to strengthen Cloud security and support Cloud revolution that enables unprecedented interconnection of networked processes operated in a blurred real and virtual world boundary.

The future work covers the development of novel verification method, as well as the building of the benchmark for evaluating the performance and the threat/risk from cyber-physical attacks.

ACKNOWLEDGMENT

This work supported by City University of Hong Kong Strategic Grant Number 7004228 is acknowledged.

REFERENCES

[1] J. A. Stankovic, "Research Directions for the Internet of Things,"Internet of Things Journal, IEEE, vol.1, pp.3-9, 2014, doi: 10.1109/JIOT.2014.2312291.

[2] B. Pranggono, Y. Yang, K. McLaughlin and S. Sezer, "Intrusion Detection System for Critical Infrastructure," The State of the Art in Intrusion Prevention and Detection, A-S. K. Pathan, Ed., ed:London, UK: CRC Press, pp.150-170, 2014.

[3] W. Aman and E. Snekkenes, "An Empirical Research on InfoSec Risk Management in IoT-based eHealth" MOBILITY 2013 : The Third International Conference on Mobile Services, Resources, and Users, 2013, pp.99-107, ISSN**:** 2308-3468, ISBN: 978-1-61208-313-1.

[4] R. Barbulescu, P. Gaudry, A. Joux, and E. Thom´e, "A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic," Advances in Cryptology— EUROCRYPT 2014, volume 8441 of LNCS, Springer, 2014, pp.1–16, doi: 10.1007/978-3-642-55220-5_1.

[5] G. Adj, A. Menezes, T. Oliveira and F. Rodr´ıguez-Henr´ıquez, "Weakness of $F_{36\cdot509}$ for discrete logarithm cryptography," Pairing-based Cryptography—Pairing 2013, volume 8365 of Lecture Notes in Computer Science, Springer, pp. 20-44, 2013.

[6] R. Granger, T. Kleinjung, and Jens Zumbrägel, "Breaking '128-bit secure' super singular binary curves," CRYPTO (2) 2014, pp. 126-145, doi:10.1007/978-3-662-44381-1_8.

[7] Hankerson D, Menezes A, and Vanstone S, Guide to elliptic curve cryptography, Springer, Berlin, 2004.

[8] D.F.Aranha, P.S.L.M. Barreto, G.C.C.F. Pereira, and J.E. Ricardini, "A note on high-security general-purpose elliptic curves," Cryptology ePrint Archive, Report 2013/647 (2013), Available from: http://eprint.iacr.org/.

[9] M. Roy1, N. Deb, and A. J. Kumar, 2014. "Point Generation And Base Point Selection In ECC," An Overview, International Journal of dvanced Research in Computer and Communication Engineering, Vol. 3, Issue 5, pp. 6711-6713, May 2014.

[10] L.M. Cheng, CW So, and L.L. Cheng. An improved forward secrecy protocol for next generation EPCGlogal tag, Development and Implementation of RFID Technology, I-Tech Education and Publishing KG, Editor Cristina Turcu , Jan. 2009, pp. 317-332, ISBN 978-3-902613-54-7.

[11] A. Juels, "Minimalist cryptography for low-cost RFID tag," Conference on Security in Communication Networks – SCN'04, LNCS, Amalfi, Italia, September (2004), Springer-Verlag, pp. 149-164, 2004.

[12] J. W. Bos, C. Costello, P. Longa, and M. Naehrig, Selecting elliptic curves for cryptography : An efficiency and security analysis, IACR Cryptology ePrint Archive, 2014:130, 2014. Available from: http://eprint.iacr.org/.

 ISBN: 978-1-61208-396-4