A Password-Authenticated Key Agreement Scheme Based on ECC Using Smart Cards

Aqeel Khalique Kuldip Singh Sandeep Sood
Department of Electronics & Computer Engineering,
Indian Institute of Technology Roorkee
Roorkee, India

ABSTRACT
Public Key Cryptography (PKC) is recently playing an essential role in electronic banking and financial transactions. Elliptic Curve Cryptography (ECC) is one of the best public key techniques for its small key size and high security and is suitable for secure access of smart cards because implementation on smart cards is challenging due to memory, bandwidth, and computation constraints. In this paper, we proposed a password-authenticated key agreement scheme based on ECC. Our scheme provides more guarantees in security as follows: 1) the computation and communication cost is very low; 2) a user can freely choose and change his own password; 3) the privacy of users can be protected; 4) it generates a session key agreed upon by the user and the server; 5) it provides both implicit key and explicit key confirmation; and 6) it can prevent the offline dictionary attack even if the secret information stored in a smart card is compromised. And yet, our scheme is simpler and more efficient for smart card authentication.

Categories and Subject Descriptors
D.4.6 [**Operating Systems**]: Security and Protection –*access controls*, a*uthentication*; E.3 [**Data**]: Data Encryption –*Public key cryptosystems.*

General Terms
Security, Authentication, ECC.

Keywords
mutual authentication, elliptic curve cryptosystem, key exchange.

1. INTRODUCTION

To access resources from a remote system, users should have proper access rights. One of the simpler and more efficient mechanisms is the use of a password authentication scheme. If a user wants to log in a remote server, he has to submit his ID and PW to the server. The remote server receives the login message and checks the eligibility of the user by referencing the password or verification table. If the submitted ID and PW match the corresponding pair stored in the server's verification table, the user will be granted access to the server. In 1981, Lamport [1] proposed the first well-known password based remote user authentication scheme without using encryption techniques. However, high hash overhead and the necessity for password resetting decrease its suitability for practical use. The Lamport scheme is not secure, due to some vulnerability. Since then, many similar schemes have been proposed, and each has its pros and cons [2][3].A remote password authentication scheme is used to authenticate the legitimacy of the remote user over an insecure channel. Through knowledge of the password, the remote user can create a valid login message to the authentication server (AS). AS check the validity of the login message to provide the access right. Two problems are found in this existing traditional mechanism.

1. The administrator of the server will come to know the password, because the server maintains the password table.
2. An intruder can impersonate a legal user by stealing the user's ID and PW from the password table.

To overcome this problem, various schemes had been proposed. Key agreement protocol is one of the fundamental cryptographic primitives after encryption and digital signature, which allows two or more communication parties to establish a secret session key over an open network. The fundamental security goals of key establishment protocols are said to be implicit key authentication and explicit key authentication [4]. Let A and B be two honest entities, i.e., legitimate entities who execute the steps of a protocol correctly. Informally speaking, a key agreement protocol is said to provide implicit key authentication (IKA) (of B to A) if entity A is assured that no other entity aside from a specifically identified second entity B can possibly learn the value of a particular secret key. A key agreement protocol which provides implicit key authentication to both participating entities is called an authenticated key agreement (AK) protocol. A key agreement protocol is said to provide key confirmation (of B to A) if entity A is assured that the second entity B actually has possession of a particular secret key. If both implicit key authentication and key confirmation (of B to A) are provided, the key establishment protocol is said to provide explicit key authentication (EKA) (of B to A). A key agreement protocol which provides explicit key authentication to both entities is called an authenticated key agreement with key confirmation (AKC) protocol [5][6].

In 1985, Koblitz [7] and Miller [8] independently proposed using the group of points on an elliptic curve defined over a finite field in discrete logarithm cryptosystems. Today elliptic curves are used widely in public key cryptosystems. We had proposed the scheme based on ECC for its greater security at smaller key length and various features favouring the properties of smart cards.

1.1 Smart Card

Smart cards are small, portable, tamper-resistant devices providing users with convenient storage and processing capability. Because of their unique form factor, smart cards are proposed for use in a wide variety of applications such as electronic commerce, identification, and health care. The majority of the smartcards on the market today have between 128 and 1024 bytes of RAM, 1 and 16 kilobytes of EEPROM, and 6 and 16 kilobytes of ROM with the traditional 8-bit CPU typically

clocked at a mere 3.57 megahertz. To be practical for widespread use, however, smart cards also need to be inexpensive. Any addition to memory or processing capacity increases the cost of each card because both are extremely cost sensitive. Smart cards are also slow transmitters, so to achieve acceptable application speeds, data elements must be small. Thus ECC is a perfect choice for smart cards for the following reasons [9]:

- **Less EEPROM and Shorter Transmission Times**
  The strength of the ECDLP algorithm means that strong security is achievable with proportionately smaller key and certificate sizes. The smaller key size in turn means that less EEPROM is required to store keys and certificates and that less data needs to be passed between the card and the application so that transmission times are shorter.
- **Scalability**
  As smart card applications require stronger security, ECC can continue to provide the security with proportionately fewer additional system resources without increasing their cost.
- **No Coprocessor**
  Other public -key systems involve so much computation that a dedicated hardware device known as a crypto coprocessor is required. The crypto coprocessors not only take up precious space, they add about 20 to 30 percent to the cost of the chip. With ECC, the algorithm can be implemented in available ROM, so no additional hardware is required to perform strong, fast authentication.
- **On Card Key Generation**
  With ECC, the time needed to generate a key pair is so short that even a device with the very limited computing power of a smart card can generate the key pair, provided a good random number generator is available.

In this paper, we had proposed a design for remote authentication key agreement scheme based on ECC using smart cards. We discuss the elliptic curve cryptography in section 2. In section 3 we present the scheme. In section 4, we show the security analysis. In section 5 we show the cost and functionality consideration among our scheme and the related schemes. Finally, we make a conclusion in Section 6.

2. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptography makes use of elliptic curves in which the variables and coefficients are all restricted to elements of a finite field. Typically, elliptic curves are defined over either the integers modulo a prime number (GF(p)) or over binary polynomials (GF(2m)). An elliptic curve is a cubic equation of the form:

$y^2 + axy + by = x^3 + cx^2 + dx + e.$ (1)

where a, b, c, d, and e are real numbers.

In an elliptic curve cryptosystem (ECC), the elliptic curve equation is defined as the form of $E_p(a, b)$:

$y^2 = x^3 + ax + b \pmod p$ (2)

over a prime finite field F_p, where $a, b \varepsilon F_p$, $p > 3$, and $4a^3 + 27b^2 \pmod p \neq 0$.

Generally, the security of ECC relies on the difficulties of the following problems [10].

- **Definition 1** Given two points P and Q over $E_p(a, b)$, the elliptic curve discrete logarithm problem (ECDLP) is to find an integer $s \varepsilon F_p^*$ such that Q = s.P.
- **Definition 2** Given three points P, s.P, and t.P over $E_p(a, b)$ for s; $t \varepsilon F_p^*$, the computational Diffie-Hellman problem (CDHP) is to find the point (s.t).P over $E_p(a, b)$.
- **Definition 3** Given two points P and Q = s.P + t.P over $E_p(a, b)$ or s; $t \varepsilon F_p^*$, the elliptic curve factorization problem (ECFP) is to find two points s. P and t.P over $E_p(a, b)$.

Up to now, there is no algorithm to be able to solve any of the above problems [10]

2.1 Finite Field

A finite field consists of a finite set of elements together with two binary operations called addition and multiplication, which satisfy certain arithmetic properties. The order of a finite field is the number of elements in the field. There exists a finite field of order q if and only if q is a prime power. If q is a prime power, then there is essentially only one finite field of order q; this field is denoted by F_q. There are, however, many ways of representing the elements of F_q. Some representations may lead to more efficient implementations of the field arithmetic in hardware or in software. If $q=p^m$ where p is a prime and m is a positive integer, then p is called the characteristic of F_q and m is called the extension degree of F_q.. More details about the prime field and finite field can be found in [11].

2.2 Elliptic Curves Operations over Finite Fields

The main operation is Point multiplication is achieved by two basic elliptic curve operations.

1. Point addition, adding two points J and K to obtain another point L i.e. L= J + K, require 1 inversion and 3 multiplication operation.

2. Point doubling, adding a point J to itself to obtain another point L i.e. L = 2J, requires 1 inversion and 4 multiplication operation.

2.2.1 Point Addition

Point addition is the addition of two points J and K on an elliptic curve to obtain another point L on the same elliptic curve.Consider two points J and K on an elliptic curve as shown in Figure 1. If K≠ -J then a line drawn through the points J and K will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of addition of points J and K. Thus on an elliptic curve L = J + K. If K = -J the line through this point intersect at a point at infinity O. Hence J + (-J) = O. A negative of a point is the reflection of that point with respect to x-axis [11].

2.2.2 Point doubling

Point doubling is the addition of a point J on the elliptic curve to itself to obtain another point L on the same elliptic curve To double a point J to get L, i.e. to find L = 2J, consider a point J on an elliptic curve as shown in Figure 2. If y coordinate of the point J is not zero then the tangent line at J will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of doubling the point J, i.e., L = 2J. If y coordinate of the

point J is zero then the tangent at this point intersects at a point at infinity O. Hence 2J = O when y_j=0. Figure 2 shows point doubling [11]

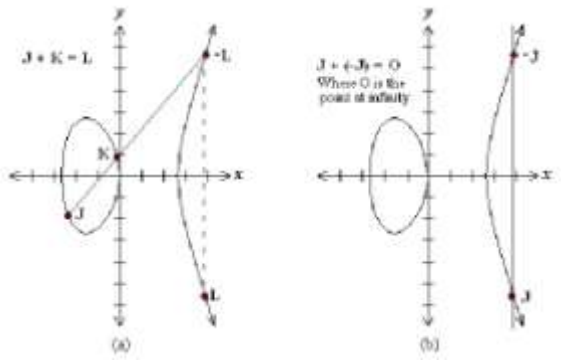

Figure 1. Point Addition

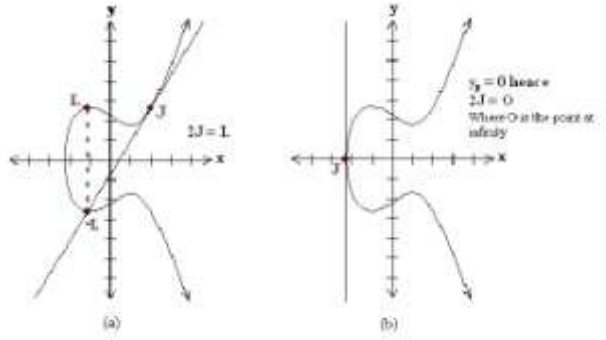

Figure 2. Point Doubling

2.2.3 Algebraic Formulae over F_p

- P+O=O+P=P for all P ε E(F_p)
- If P=(x, y) ε E(F_p) then (x, y)+(x,-y)=O. (The point (x,-y) is denoted by –P, and is called the negative of P, observe that –P is indeed a point on the curve.
- Point addition Let P=(x_1, y_1) ε E(F_p) and Q=(x_2, y_2) ε E(F_p) , where P≠ ± Q. Then P+Q=(x_3, y_3) where
  $x_3= \{(y_2-y_1)/(x_2-x_1)\}^2 - x_1-x_2$ and $y_3= \{(y_2-y_1)/(x_2-x_1)\}(x_1-x_3) -y_1$
- Point doubling Let P=(x_1, y_1) ε E(F_p) where P≠ -P. Then 2P=(x_3, y_3) where
  $x_3=\{(3x_1^2+a)/2y_1\}^2-2x_1$ and $y_3=\{(3x_1^2+a)/2y_1\}^2 (x_1-x_3) -y_1$

2.2.4 Algebraic Formulae over F2m

- P+O=O+P=P for all P ε E(F_2^m)
- If P=(x, y) ε E(F_p) then (x, y)+(x, -y)=O. (The point (x, -y) is denoted by –P, and is called the negative of P, observe that –P is indeed a point on the curve.
- (Point addition) Let P=(x_1, y_1) ε E(F_2^m) and Q=(x_2, y_2) ε E(F_2^m) , where P≠ ± Q. Then P+Q=(x_3,y_3) where
  $x_3= \{(y_2+y_1)/(x_2+x_1)\}^2 + \{(y_2+y_1)/(x_2+x_1)\}+ x_1 + x_2 +a$
  and $y_3= \{(y_2+y_1)/(x_2+x_1)\}(x_1+x_3) +x_3 + y_1$
- (Point doubling) Let P=(x_1,y_1) ε E(F_2^m) where P≠ -P. Then 2P=(x_3,y_3)
  where $x_3=x_1^2+(b/ x_1^2)$ and $y_3=x_1^2+\{x_1+(y_1/x_1)\}x_3 + x_3$

3. PROPOSED SCHEME

The proposed scheme has been divided into 4 phases: **parameter generation phase, registration phase, authentication phase, and password change phase**.

A. In **parameter generation phase**, AS chooses an elliptic curve E over a finite field F_p such that the discrete logarithm problem is hard in E(F_p). The set of all the points on E is denoted by E(F_p). AS also chooses a point G ∈ E(F_p) such that the subgroup generated by has a large order n. AS publishes the parameters (p, E, G, n).

B. In **registration phase**, there is a unique identifier, ID associated to each user. AS generates V = h(ID||K_s) ⊕ h(PW) and IM = E_{Ks}(ID||r) where PW is initial password selected by the AS, r is a random number to provide identity security and K_S is the private key of AS. AS determines the initial password for U. After receiving the smart card, U is able to immediately change the initial password.

C. In **authentication phase**, a session key K_{SU} is established. The steps are shown below in Figure 3.

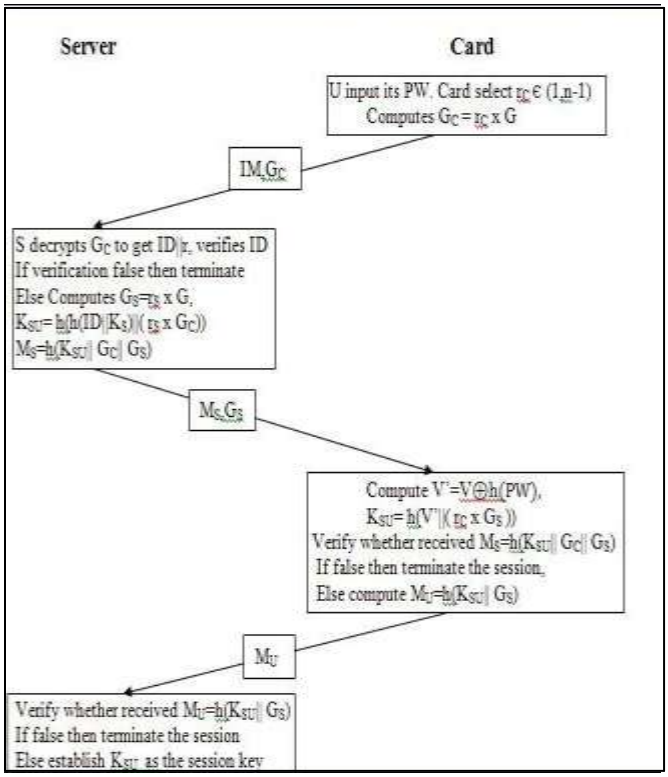

Figure. 3 Authentication phase of proposed scheme

D. In **Password-Change Phase w**hen U wants to change his password PW with a new one, U enters the old password PW, and requests to change password. Then, U enters the new password PW^*. U's smart card computes $V^* = V \oplus h(PW) \oplus h(PW^*)$, which yields h(ID||KS) ⊕ h(PW^*), and then replaces V with V^*. U can freely change his password and reduces the possibility of the insider attack.

4. SECURITY ANALYSIS

In this section, we will analyze the security of our proposed scheme. The main assumption for guarantee of security lies in:

1) The elliptic-curve Diffie–Hellman problem is hard;

2) The hash function h ( ) is the pseudorandom permutation for key derivation;

Our scheme can achieve the goal of user authentication and key agreement with great assurance and certainly can prevent the well-known attacks, such as the replay, parallel session, reflection, interleaving, and man-in-the-middle attacks.

Identity Protection: The identifier ID is never explicitly transmitted via the insecure channel. Therefore, both schemes can provide the user's identity protection. Even if the smart card is lost the attacker cannot get the identifier ID in our improved scheme, because he cannot derive the identifier ID from the parameters V and IM without the master secret key K_S.

Replay attack: The replay attack is when an attacker tries to imitate the user to log in to the server by resending the messages transmitted between the user and the server. In our scheme, we use nonces to prevent this kind of attack. In our proposed scheme, the smart card chooses a nonce r_C and computes $G_C = (r_C \times G)$ and then sends it to the server. The second nonce r_S is selected by the server and server computes $G_S = (r_S \times G)$.

Passive attack: A passive attack can be possible if C, the attacker, make a guess at the session key using only information obtainable over network. If the attacker C performs a passive attack, then the session will terminate with both parties accepting. That is, B and A successfully identify themselves to each other, and they both compute the session key. So, C the adversary cannot compute any information about the common shared session key K_S due to the intractability of elliptic curve discrete logarithm problem. Therefore the proposed scheme resists against the passive attack.

Dictionary attack: The dictionary attack could be performed in offline or online mode. An on-line password guessing attack cannot succeed since AS can limit the number of attempts. On the other hand, the offline dictionary attack is very powerful since the attacker does not need to interact with the legitimate entities and can use a lot of computing power. The messages $\{IM, G_C\}$, $\{M_S, G_S\}$, and $\{M_U\}$ of a legitimate authentication session and U's parameter V cannot help the attacker to verify the guessed password, because the corresponding value $r_S \times r_C \times G$ is not available. So the proposed scheme can prevent both type of dictionary attacks.

Smart Card Loss Attack: Suppose user loses his smart card, the adversary cannot use this card without knowing the password of the user. Suppose an adversary wants to change the password, he must know the original password. Thus his attempt to impersonate user fails.

Parallel Session Attack: Suppose an adversary intercepts the message $\{IM, G_C\}$, $\{M_S, G_S\}$, and $\{M_U\}$ to create a valid login. But he cannot succeed as G_C and G_S depends on random r_C and r_S. The adversary cannot find the value of r_C and r_S due to the intractability of elliptic curve discrete logarithm problem.

Explicit Key Confirmation: Using three exchanged messages in the authentication phase, our scheme achieved the explicit key confirmation. AS needs the correct session key K_{SU} to generate the value M_S, which is equal to $h(K_{SU}||G_C||G_S)$. Therefore, AS can be assured that U has actually computed $K_{SU} = h(V^*||(r_C \times G_S))$, after AS has verified that the value M_U is equal to $h(KSU||G_S)$ and thus, U can be assured that AS has actually computed $K_{SU} = h(h(ID||K_S)||(r_S \times G_C)$.

5. COST AND FUNCTIONALITY

We had made some assumption to do comparison analysis with the scheme of Juang *et al*. [12]. Table 1 summarizes the security functionalities that are believed to be provided by the Juang *et al.* scheme and our scheme.

Table 1. Functionality

	Juang *et al.* scheme	**Proposed scheme**
Password Table	Not required	Not required
Password	Provided during registration	Provided during registration
Implicit Key Confirmation	Yes	Yes
Explicit Key Confirmation	No	Yes
Verification Table	Required	Required but of smaller size

In the storage cost concern, our scheme requires the smart card to store the parameters V and IM instead of the parameters V , IM, ID, CI, and b in the scheme of Juang *et al.* We can further estimate that the parameters V , IM, ID, CI, and b in the scheme of Juang *et al.* need 128 + 256 + 32 + 32 + 64 = 512 bits of storage space. Correspondingly, the parameters V and IM in our improved scheme need 128 + 128 = 256 bits of storage space. AS need a 128-bit storage space for the secret parameter K_S in our scheme. In the scheme of Juang *et al.*, AS need about 163 + 128 = 291 bits of storage space for the secret parameters x and K_S. We list the storage costs of the scheme of Juang *et al.* and our scheme in Table 2.

Table 2. Storage Cost

	Juang *et al.* scheme	**Proposed scheme**
Smart Card	512 bits	256 bits
Server	291 bits (two secret keys x and K_S)	128 bits (only one secret key K_S)

Consider the communication costs. In a normal session run, our improved scheme needs exchange data IM, G_C, M_S, G_S, and M_U, while that of Juang *et al*. need exchange data IM, $E_{V(e)}$, N_S, M_S, and M_U. Let the nonce N_S be 128 bits. The communication cost of a normal session run is 128 + 326 + 128 + 326 + 128 = 1036 bits in our improved scheme and 384 + 384+ 64 + 128 + 128 =

1088 bits in that of Juang *et al*. For the password-changing operation, Juang *et al*. need to exchange data E_{KSU} (ID, h ($PW^{*}||b^{*}$)) and E_{KSU} (IM^{*}) costing 512 bits while our scheme need not exchange any data. In Table 3, we show the communication costs of the scheme of Juang *et al*. and our scheme.

Table 3. Communication Cost

	Juang *et al.* scheme	**Proposed scheme**
Log in	1088 bits	1036 bits
Password Change	512 bits	NIL

In Table 4 and 5, we tabulate the computation costs in the phases of the schemes. The names of the computation operations have been abbreviated: H denotes the cryptographic hash computation; E denotes the symmetric encryption or decryption computation, and E_M denotes the scalar multiplication computation over the elliptic curve. Consider the computation cost of the smart card. We can see that our scheme is a little more efficient than the scheme of Juang *et al*. Due to the limited hardware resources; the smart card is always unable to provide powerful computation capability. Hence, it is a desirable feature.

Table 4. Computation Cost of Juang *et al*. scheme

Entity / Phases	**Smart Card**	**Server**	**Total**
Parameter Generation	-	$1E_M$	$1E_M$
Registration	1H	3H + 1E	4H + 1E
Pre computation	$2E_M$	-	$2E_M$
Log in	3H + 1E	$1\ E_M$ + 4H + 1E	$1E_M$ + 7H +2E
Password Changing	1H + 2E	3E + 1H	2H + 5E

Table 5. Computation Cost of proposed scheme

Entity / Phases	**Smart Card**	**Server**	**Total**
Parameter Generation	-	-	-
Registration	-	1E + 2H	1E + 2H
Log in	$2E_M$ + 4H	$2E_M$ + 4H + 1E	$4E_M$ + 8H + 1E
Password Changing	3H	-	3H

6. CONCLUSION

In this paper, we proposed an authentication scheme which is secure and simpler than existing schemes based on ECC using smart cards. Our scheme has low communication and computation cost by using elliptic curve cryptosystems and can prevent the insider attack. Smart card are constrained devices having less memory and low processing power, we had shown functionality analysis which itself favors the proposed scheme. Our proposed scheme is very useful in limited computation and communication resource environments to access remote information systems. Our scheme is secure as it inherits the security and implementation properties of the elliptic curve cryptosystems, which seem to offer the highest cryptographic strength per bit among all existing public-key cryptosystems and even provides more desirable properties. Therefore, our scheme is more practical than the previous related schemes for smart cards.

7. REFERENCES

[1] Lamport, L. 1981. Password authentication with insecure communication. Communication of the ACM. vol. 24, No. 11. 770-772.

[2] Mangipudi, K., Katti, R. 2006. A Hash based Strong Password Authentication Protocol with User Anonymity. International Journal of Network Security. vol. 2. No.3. 205-209

[3] Jena, D., Jena, S. K., Mohanty, D., Panigrahy, S. K. 2008. A Novel Remote User Authentication Scheme using Smart Card based on ECDLP. International Conference on Advanced Computer Control.

[4] Song, B., Kim, K. 2000. Two-Pass Authenticated Key Agreement Protocol with Key Confirmation. Progress in Cryptology. INDOCRYPT 2000. LNCS 1977, Springer-Verlag. 58-65.

[5] Wilson, S. B., Menezes, A. 1999. Authenticated Diffie-Hellman Key Agreement Protocols. Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC '98). LNCS 1556, Springer-Verlag. 339-361.

[6] Menezes, A., Oorschot, P. V., Vanstone, S. 1997. Handbook of Applied Cryptography. CRC Press.

[7] Koblitz, N., 1987. Elliptic curve cryptosystems. Mathematics of Computation 48, 203-209.

[8] Miller, V., 1985. Use of elliptic curves in cryptography. CRYPTO 85.

[9] The Elliptic Curve Cryptosystem for Smart Cards. 1998. A Certicom White Paper.

[10] Li, F., Xin, X., Hu, Y. 2008. Identity-based broadcast signcryption. Computer Standard and Interfaces. vol 30. 89-94.

[11] Hankerson, D., Menezes, A., Vanstone, S., 2004. Guide to Elliptic Curve Cryptography. Springer.

[12] Juang, W. S., Chen, S. T., Liaw, H.T. 2008. Robust and Efficient Password-Authenticated Key Agreement Using Smart Cards. IEEE Transactions on Industrial Electronics. vol. 55, No. 6.