

A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications

Pierre E. ABI-CHAR, Abdallah MHAMED
UMR CNRS 5157
GET/Institut National des Télécommunications
9 rue C. Fourier - 91011 Evry CEDEX - France
{pierre.abichar; abdallah.mhamed}@int-edu.eu

Bachar EL-HASSAN
Libanese University
Faculty of Engineering
Tripoli - Lebanon
bachar_elhassan@ul.edu.lb

Abstract

The increasing progress in wireless mobile communication has attracted an important amount of attention on the security issue. To provide secure communication for mobile devices, authenticated key agreement protocol is an important primitive for establishing session key. So far, several protocols have been proposed to provide robust mutual authentication and key establishment for wireless local area network (WLAN). In this paper we present a fast and Secure Authenticated Key Agreement (EC-SAKA) protocol based on Elliptic Curve Cryptography. Our proposed protocol provides secure mutual authentication, key establishment and key confirmation over an untrusted network. The new protocol achieves many of the required security and performance properties. It can resist dictionary attacks mounted by either passive or active networks intruders. It can resist Man-In-The Middle attack. It also offers perfect forward secrecy which protects past sessions and passwords against future compromise. In addition, it can resist known-key and resilience to server attack. Our proposed protocol uses ElGamal signature techniques (ECEGS). We show that our protocol meets the above security attributes under the assumption that the elliptic curve discrete logarithm problem is secure. Our proposed protocol offers significantly improved performance in computational and communication load over comparably many authenticated key agreement protocols such as B-SPEKE, SRP, AMP, PAK-RY, PAK-X, SKA, LR-AKE and EC-SRP.

1 Introduction

In key agreement protocol two or more distributed entities need to share some key in secret, called session key. This session key can then be used to achieve some cryptographic goal such as confidential communication chan-

nel between entities or data integrity. There are two kinds of key establishment protocols: Key transport protocols in which a key is created by one entity and securely transmitted to the second entity, and Key agreement protocols in which both parties contribute information which jointly establish the shared key [14]. A key agreement protocol is said to provide implicit key authentication 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 entities is called an authenticated key agreement protocol. If both implicit key authentication and key confirmation are provided, then the key establishment protocol is said to provide explicit key authentication. A key agreement protocol which provides explicit key authentication to both entities is called an authenticated key agreement with key confirmation [14]. The security of Elliptic Curve cryptography relies on the discrete logarithm problem over the points on an elliptic curve. The best known methods to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) are Pollard approach and Pohlig-Hellman method. They are fully exponential while the best known methods to solve the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP), on which most of the non-ECC cryptosystems rely, are sub-exponential. In fact, ECC can significantly reduce the computation and storage overhead.

In this paper we present a fast and secure three-pass authenticated key establishment protocol for low power mobile wireless devices that provides secure mutual authentication and key agreement with key confirmation. The EC-SAKA (Secure Authenticated Key Agreement) is based on the Elliptic Curve Cryptography [19], on the EC ElGamal Signature Scheme (ECEGS), on SKA (Simple Key Agreement) protocol [17] and on the assumption that the ECC discrete logarithm problem is secure [19]. Our proposed protocol achieves many of desirable security requirements

and performances.

The ECEGS is used to make the proposed protocol fast and efficient, both from a computational point of view and in the amount of information that needs to be exchanged in the scheme. It is also used to minimize the amount of computational performed by Alice, in particular. This is desirable because, in many practical applications, Alice's computations will be performed by a smart card with low computing power, while Bob's computations will be performed by a more powerful computer.

The protocol described in this paper establish a shared key K between the two entities. A key derivation function should then be used to derive the session key K_s based on the shared key.

The remainder of this paper is organized as follows. Section 2 reviews the desirable properties needed for WLAN (Wireless Local Area Network) authenticated protocols. Section 3 presents the overall architecture of our proposed protocol. In section 4, the security analysis is described. In section 5, a complete comparison over comparably many protocols is listed. Finally, section 6 makes concluding remarks.

2 DESIRABLE PROPERTIES FOR KEY AGREEMENT PROTOCOLS

A number of desirable properties for key agreement protocols have been identified [2] and nowadays most of the protocols are analyzed using these properties which are described below:

-Known-key security: Each run of a key agreement protocol between two entities A and B should produce a unique shared secret key called session key K_s . A protocol should still achieve its goal in the face of an adversary who has learned some other session key.

-Perfect forward secrecy: If long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities is not affected.

-Key-compromise impersonation: Suppose that A's long-term private key is disclosed. Clearly an adversary that knows this value can now impersonate A, since it is precisely this value that identifies A. However, it may be desirable that this loss does not enable an adversary to impersonate other entities to A.

-Unknown key-share: Entity A cannot be coerced into sharing a key with entity B without A's knowledge, i.e., when A believes the key is shared with some entity $C \neq B$, and B (correctly) believes the key is shared with A.

-Key control: No other entity should be able to force the session key to a preselected value.

In addition, Identification protocols should have other properties which are related to performance. Because round

trips and large blocks are critical factors in terms of communication load and because exponentiations and random numbers are to be critical factors in terms of computation load, such properties are listed below:

-Computational efficiency: this includes the number of operations required to execute a protocol. In order to achieve this property, the protocol should have the minimum number of operation as possible.

-Communication efficiency: This includes the number of passes (message exchanges) and the bandwidth required (total number of bits transmitted).

Other desirable properties are:

-Nature of security guarantees: including provable security and zero-knowledge properties.

-Storage of secrets: This refer to the location and the method used (e.g., software only, local disks, hardware tokens, etc.) to store critical keying material.

3 Our proposed Protocol

In this section we describe the EC-SAKA protocol in which two entities are both proving their identities to each other and establish a common session key in order to elaborate a secure connection. Alice and Bob represent a client and a server respectively.

3.1 EC-SAKA Protocol Parameters

Many researchers have examined elliptic curve cryptosystems, which were firstly proposed by Miller [15] and Koblitz [8]. The elliptic curves which are based on the elliptic curve discrete logarithm problem over a finite field have some advantages than other systems: the key size can be much smaller than the other schemes since only exponential-time attacks have been known so far if the curve is carefully chosen [9], and the elliptic curve discrete logarithms might be still intractable even if factoring and the multiplicative group discrete logarithm are broken. In this paper we use an elliptic curve E defined over a finite field F_p . The elliptic curve parameters to be selected [12] and [11] are:

1 -Two field elements a and $b \in F_p$, which define the equation of the elliptic curve E over F_p (i.e., $y^2 = x^3 + ax + b$ in the case $p \geq 4$, where $4a^3 + 27b^2 \neq 0$).

2 -Two field elements x_p and y_p in F_p , which define a finite point $P(x_p, y_p)$ of prime order in $E(F_p)$ (P is not equal to O , where O denotes the point at infinity).

3 -The order n of the point P .

The Elliptic Curve domain parameter can be verified to meet the following requirements [12] and [11]. In order to avoid the Pollard-rho [16] and Pohling-Hellman algorithms for the elliptic curve discrete logarithm problem,

it is necessary that the number of F_p -rational points on E , denoted by $\#E(F_p)$, be divisible by a sufficiently large prime n . To avoid the reduction algorithms of Menezes, Okamoto and Vanstone [1] and Frey and Ruck [5], the curve should be non-supersingular (i.e., p should not divide $(p + 1 - \#E(F_p))$). To avoid the attack of Semaev [18] on F_p -anomalous curves, the curve should not be F_p -anomalous (i.e., $\#E(F_p) \neq p$).

The table below (Table 1) shows the ECC mathematical parameters that are used in our proposed protocol.

Table 1. EC Mathematical Notations

Index	Explanation
$h()$	one-way hash function
p, q	large prime numbers, where $p = 2 \cdot q + 1$
P, Q	Random points over elliptic curve
a, b	Random generated private keys
A, B	Agreed public key
E	non-supersingular elliptic curve
B	$B \in E(F_q)$ with order q
$x(Q)$	x coordinate of point Q
x_u	user 'u' secret key
Y_u	u 's public key with $Y_u = (x_u B) \text{ mod } p$
pw	The user 's' password

In the following, we will give an introduction to the EC-discrete logarithm problem, to Diffie-Hellman key exchange based on EC and to the elliptic curve based ElGamal signature scheme (ECEGS), which are the core theories used in our proposed protocol.

Let E be an elliptic curve defined over a finite field F_p and let $P \in E(F_p)$ be a point of order n . Given Q where $Q \in E(F_q)$, the elliptic curve discrete logarithm problem is to find the integer l , $0 \leq l \leq n - 1$, such that $Q = l \cdot P$.

The Diffie-Hellman key agreement protocol runs as follows: Alice selects a random number n_a and computes $Y_a = n_a B$, he sends Y_a to Bob. Similarly Bob computes $Y_b = n_b B$ and sends Y_b to Alice. Alice and Bob generate the same key $K = n_a Y_b B = n_b Y_a = n_a n_b B$.

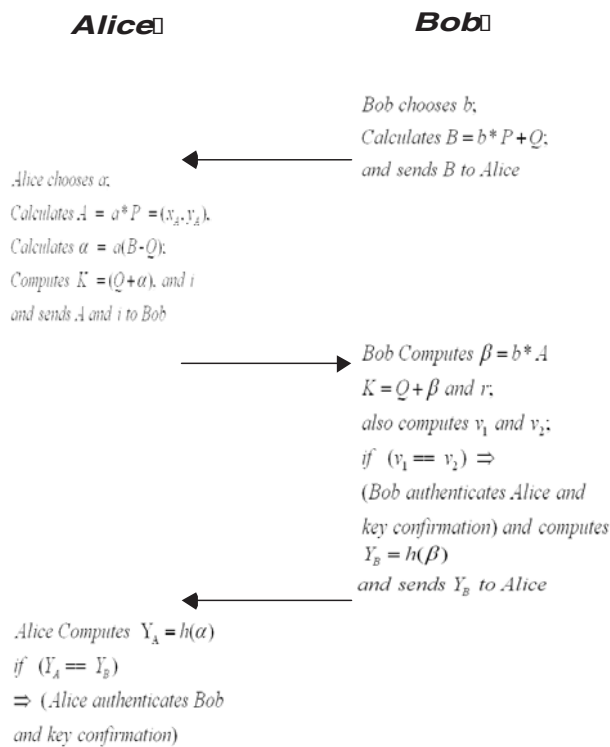
The ECEGS runs as follows: Alice selects a random number x_a , where $2 \leq x_a \leq n - 2$, as his secret key and computes the corresponding public key $Y_a = x_a B$. Therefore the public key and the private key are (E, Y_a, B, n) and x_a . To generate a signature for a message m , Alice will select a random number k , where $2 \leq k \leq n - 2$ computes $R = kB$ and computes $r = x(KB) \text{ mod } n$. If $r \neq 0$, then computes $s = K^{-1}(h(m) + x_a r) \text{ mod } n$. The couple (R, s)

will be Alice's signature of m . To verify the signature Bob will first confirm that r and $s \in [2, n-2]$ and then computes $v_1 = sR$ and $v_2 = h(m)B + rY_a$. Finally Bob will accept the signature if and only if $(v_1 == v_2)$.

3.2 EC-SAKA Protocol Description

The main goal of our proposed protocol is to achieve mutual authentication and key confirmation between the client and the server in order to establish a secure channel. Our proposed applies the ECEGS to the SKA protocol to enhance the safely level and protocol simplification in terms of computational and communications load. Our proposed protocol consists of three flows and it is illustrated in Figure 1.

Figure 1. The EC-SAKA Protocol Scheme



Before running the authentication procedure, the client select an elliptic curve $E(Z_p)$ defined on Z_p . Alice, select a random point P over the elliptic curve with order n . n is a large prime number. Alice chooses a password pw , computes $x = h(pw)$ and calculates Q where $Q = x * P$. In addition, Alice generates strong number p and q where $p = 2 * q + 1$. Once the following parameters (E, Q, P, p, q, pw) are generated, Alice transfers (E, Q, P, n) to the server, Bob, in a secure way. Once

these steps are done, the session key generation procedure will be executed as follow:

Within the first flow, Bob chooses a random challenge b , where $1 \leq b \leq n - 1$, then he calculates the point B where

$$B = b * P + Q \quad (1)$$

Finally he sends B to Alice.

Within the second flow, Alice chooses a random challenge a , where $1 \leq a \leq n - 1$, then computes A where

$$A = a * P = (x_A, y_A) \quad (2)$$

and calculates α where

$$\alpha = a(B - Q) \quad (3)$$

and $K = Q + \alpha$. In addition, Alice calculates $r = (x_A \bmod n)$ and checks that $r \neq 0$. If so, then Alice computes

$$i = a^{-1}(h(\alpha) + x * r) \bmod(n) \quad (4)$$

Finally (A, i) becomes the signatures pair and Alice transfers A and i to the server.

Within the third flow, Bob computes

$$\beta = b * A \quad (5)$$

Computes $K = Q + \beta$ and computes $r = x_A \bmod n$. To verify the signature, first Bob checks that $(r, i) \in [2, n - 2]$. If so, then Bob computes

$$v_1 = i * A \quad (6)$$

and calculates

$$v_2 = (h(\beta)P) + r * Q \quad (7)$$

Finally, Bob checks if $(v_1 == v_2)$, if so, Bob authenticates Alice and Bob can be confirmed that Alice has actually established the same shared session key. Then Bob computes:

$$Y_B = h(\beta) \quad (8)$$

and finally he sends Y_B to Alice.

In order to authenticate Bob, Alice will compute:

$$Y_A = h(\alpha) \quad (9)$$

and then Alice will verify the value of Y_A by checking that $(Y_A == Y_B)$, if so, if they match, then Alice authenticates Bob and Alice can be confirmed that Bob has actually established the same shared session key with her.

Finally, Alice and Bob agree on the common session key K_s where

$$K_s = h(ID(Alice)||ID(Bob)||K) \quad (10)$$

Both sides will agree on the session Key K_s if all steps are executed correctly. Once the protocol run completes successfully, both parties may use K_s to encrypt subsequent session traffic in order to create a confidential communication channel.

4 Security Analysis

In the following section, we will analyze the security of our proposed protocol.

The EC-SAKA protocol is considered to be a secure authenticated key establishment protocol, if it satisfies the following properties:

-Passive attack: Suppose that Oscar the attacker perform a passive attack, then the session will terminate with both parties accepting. That is, Bob and Alice successfully identify themselves to each other, and they both compute the session key. So, Oscar, the adversary, cannot compute any information about the common shared session key K_s by assuming the intractability of the elliptic curve discrete algorithms problem. Therefore the EC-SAKA protocol resists against the passive attack.

-Man in the middle attack (or active attack): Suppose that an attacker, Oscar, intercepts B and replaces it with B' , Oscar then receives A and i from Alice. He would like to replace i by i' , as before. However, this means that he must calculate α where $\alpha = a(B' - Q)$ but unfortunately for Oscar, he can not compute the value of α because he does not know the the value of Q neither the value of a . So, Oscar will not be able to compute K , neither K_s . Therefore the EC-SAKA protocol thwarts the man in-the-middle attack.

-Dictionary attack: In dictionary attack, the attacker finds the real password by repeating a process of guessing the password of legal client and applying the passwords. The dictionary could be performed in offline or online mode. In our proposed protocol, it is impossible to get the real password since a one way hash function is applied to the password and during the protocol process, the shared Key K used in the calculation of K_s is calculated from b and a which are generated every new session; and by assuming the intactability of elliptic curve discrete logarithm problem Therefore the EC-SAKA protocol thwarts the offline and online dictionary attack.

-Known-key attack: In this attack, an adversary will capture the session key from an eavesdropped session. In our proposed protocol, the client and the server both generates new b and a every new session, and in addition the shared key K is generated with every new session also. Thus our proposed protocol is secure against known key attacks assuming that the elliptic discrete problem is intractable.

-Perfect forward secrecy: The perfect forward secrecy is that an exposed password does not enable an attacker to derive session keys of past communication sessions. In our protocol, the security of perfect forward secrecy is based upon the assumption that the elliptic curve discrete problem is intractable and on the value of the key K . Even if the attacker knew the correct password, the attacker still cannot compute the previous session keys because K_s is derived from the shared key K which is generated from the value of a and b . Therefore, the EC-SAKA protocol satisfies the property of perfect forward secrecy.

-Resilience to server compromised: if the host's password file is compromised, an adversary can not use it to impersonate legitimate user since the server does not store the file of the password, instead the value of Q where $Q = x * P$ is stored. Thus, Oscar has to solve the elliptic curve discrete logarithm problem in order to retrieve the value of password. The EC-SAKA protocol provides resilience to server compromise.

5 Performance evaluation: Efficiency and Comparison:

Computation cost and communication cost are the most important aspects of password authentication protocols which affect the overall performance. They include number of steps, exponentiations, large blocks, symmetric encryption and decryption, hash functions and random numbers. In this section, we compare the EC-SAKA protocol with the following protocols: Leakage-Resilient Authenticated Key Exchange (LR-AKE) protocol, Simple Key Agreement (SKA) protocol, Secure Remote Password (SRP) protocol, EC-SRP, Simple Password Exponential Key Exchange (B-SPEKE) protocol, Password-Authenticated Key Exchange (PAK-X and PAK-RY) protocols and Authentication Memorable Password (AMP) protocol. The comparison is done in terms of number of steps, random numbers, exponentiations and hash functions. Table 2 shows the compared result for number of steps and exponentiation. Table 3 shows the compared result for random numbers and hash functions numbers.

It is clear from Table 2 that the EC-SAKA protocol has the minimal cost in terms of number of steps and exponenti-

Table 2. Comparison of Performance-1-

Protocol	Rounds	Exponentiations		
		Client	Server	Total
B-SPEKE	4	3	4	7
SRP	4	3	3	6
AMP	4	2	3	5
PAK-RY	3	5	4	9
PAK-X	3	5	4	9
SKA	3	2	3	5
LR-AKE	3	3	2	5
AKEECC	4	2	2	4
EC-SRP	3	2	2	4
EC-SAKA	3	1	0	1

ations compared with these above protocols. We can easily notice that B-SPEKE [4], SRP [20], AKEECC [7] and AMP [10] require 4 rounds while PAK-RY [3], PAK-X [13], SKA [17], LR-AKE [6] and EC-SAKA require 3 rounds. In addition, the computational load was clearly improved using EC-SAKA protocol because, as noted in table 2, EC-SAKA requires 1 exponentiations, one for the client and nothing for the server. While for the other protocols, including SKA, LR-AKE, AKEECC [7] and EC-SRP [7], they require at least 4 exponentiations.

Table 3. Comparison of Performance-2-

Protocol	Random N.	Hash Function N.
SRP	2	6
AMP	2	9
PAK-RY	3	8
PAK-X	3	10
SKA	2	7
LR-AKE	2/4	6
AKEECC	2	6
EC-SRP	3	5
EC-SAKA	2	5

From Table 3, we can easily notice that the EC-SAKA protocol requires 2 random numbers and 5 hash functions while all the other protocols require more. In addition, for the EC-SRP and AKEECC protocols described in [7], we can easily notice that our protocol is better then these two protocols in terms of hash functions numbers. For the EC-SRP protocols described in [7], EC-SRP protocol was proposed for a one way authentication while our proposed protocol, EC-SAKA, provides mutual authentication.

6 Conclusion

Wireless network access are nowadays very important for users. This paper describes a new network access mech-

anism for wireless local area networks. In this paper, we introduce a fast and secure authenticated key agreement protocol based on elliptic curve cryptography and provides mutual authentication and explicit key establishment. We also give a formal approach to prove its security based on the previous work by [7]. Our scheme is simple, easy to realize, and secure against both passive and active attacks. It also resists many others attacks as described in section 4. Our proposed protocol is compared to well-known protocols such as B-SPEKE, SRP, EC-SRP, AKEECC, PAK-RY, PAK-X, AMP, SKA and LR-AKE in terms of communication and computation cost and the results were well discussed in the previous section .

Acknowledgments

The authors would like to thanks the following departments RST-INT Evry and FOE-UL for their support and comments. Their suggestions and observations were extremely helpful throughout this paper.

References

- [1] T. O. A. Menezes and S. Vanstane. Reducing elliptic curve logarithms in a finite field. *IEEE Transactions on Information Theory*, vol. 39, pages 1639 – 1646, 1993.
- [2] S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. In *Proc. of Sixth IMA International Conference on Cryptography and Coding*, pages 30 – 45. Cirencester, UK, 1997.
- [3] V. Boyko, P. Mackenzie, and S. Patel. Provably secure password authenticated key exchange using diffie-hellman. *EuroCrypt*, pages 156 – 171, 2000.
- [4] D. Jablon. Extended password key exchange protocols immune to dictionary attack. *WETICE Workshop*, pages 248 – 255, 1997.
- [5] G. Frey and H. Ruck. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, vol 62, pages 865 – 874, 1994.
- [6] I. Hideki, S. Seonghan, and K. Kobara. Authenticated key exchange for wireless security. *IEEE Wireless Communications and Networking Conference*, pages 1180 – 1186, 2005.
- [7] J. K. K. Jung and T. Chung. Password-based independent authentication and key exchange protocol. *Proc. of ICICS-PCM 2003, Singapore*, 2003.
- [8] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, vol 48., pages 203 – 209, 1987.
- [9] N. Koblitz. Cm-curves with good cryptography properties. *Proc. of Crypto' 91, Santa Barbara, USA*, 1992.
- [10] T. Kwon. Ultimate solution to authenticate via memorable password. *Contribution to the IEEE P 1363 Study group for Future PKC Standards*, available for <http://grouper.ieee.org/groups/1363/>.
- [11] M. Q. J. S. L. Law, A. Menezes and S. Vanstane. An efficient protocol for authenticated key agreement. In *Designs, Codes and Cryptography*, vol. 28.
- [12] M. Q. J. S. L. Law, A. Menezes and S. Vanstane. An efficient protocol for authenticated key agreement. *Technical report CORR98-05, Department of CO, University of Waterloo*, 1998.
- [13] P. Mackenzie. More efficient password authenticated key exchange. *CT-RSA*, pages 361 – 377, 2001.
- [14] A. Menezes, P. Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2nd edition, 1996.
- [15] V. Miller. Uses of elliptic curves in cryptography. In *Proc. of Crypto '85, Santa Barbara, USA*, pages 417 – 426, 1986.
- [16] J. Pollard. Monte carlo methods for index computation mod p. *Mathematics of Computation*, vol. 32, pages 918 – 924, 1978.
- [17] E. Ryu, K. Kim, and K. Yoo. A simple key agreement protocol. In *Proc. of IEEE 37th Annual 2003 International Carrihan Conference*, pages 128 – 131, 2003.
- [18] I. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation*, vol. 67, pages 353 – 356, 1998.
- [19] D. R. Stinson. *Cryptography Theory and Practice*. Chapman and Hall/CRC, third edition, 2006.
- [20] T. Wu. Secure remote password protocol. *Interent Symposium on Network and Distribution System Security*, 1998.