Annals of University of Craiova, Math. Comp. Sci. Ser.
Volume 30(2), 2003, Pages 59–65
ISSN: 1223-6934

The agreement of the common key

Nicolae Constantinescu

Abstract. In this paper we propose a protocol for authenticated key agreement between parties of a computer network. We consider a public-key encryption scheme for authentication and an elliptic-curves-system-based for key-agreement. The presented protocol needs two communication rounds between involved parties and its security is based on the elliptic curve discreet logarithm problem, and it is made with respect to all types of attack and security discussions presented in [8, 9, 21, 22, 27]. Also, it is discussed the protocol security level and the proof of security is given.

2000 Mathematics Subject Classification. 11T71,94A62,14G50.
Key words and phrases. authentication key agreement, elliptic curve, public-key based encryption.

1. Introduction

The idea of this paper comes from efforts to confront problems in a real situation presented in a computer network. There was needed a secure key agreement between all pairs in this network without the possibility, for other parties, to get messages than are not addressed to them. It is necessary for all to be unable to substitute the identity of other parties in order to communicate under other identity. In this paper we consider the case of a pair in a computer network, let's say (for notation convenience) *Alice* and *Bob*, noted as A and B, in which each of them has a secret key from the public-key cryptosystem and an ID ($Code\ Identification$).

The public key for every participant in the network is stored in a trusted server. We consider also the case in which an adversary (someone who want to make an illegal access to information between parties or to delete, inject or modify the messages) has the opportunity to receive all the messages from the communication channel in the network and to make a passive attack or to boot an active attack. Also, our goal for the presented protocol is to implement it in a no high performance computer. As in [3, 5, 19, 20, 25], there was created a key agreement system which preserves the AK (*Authenticated Key agreement*) attributes.

The protocol was created with respect to all types of attack presented in [12] which almost all of them are discussed in section 4 of this paper. In [18] the authors describe a protocol in two steps, similar to ours, but in every step of our protocol the amount of computation is reduced, which is a major requirement in a real network.

2. Required Attributes of an Agreement Key Protocol

2.1. Authentication requirement. There where created many protocols for the *AK-problem*, based on Diffie-Hellman protocol [11], and in [22] it was presented how

Received: 16 October 2003.

is possible to break that protocol, with the well-known man-in-the-middle-attack. In order to construct a secure key exchange between A and B it is necessary to make an authentication of parties. If A wants to have a communication with B, based on symmetric encryption, they must agree a new key for this communication session (key-session). This key should be constructed with the knowledge of both of them, without possibility for anyone of them to predetermine the key or a fraction of it. The newest algorithm that fulfills all these requirements is presented in [20].

2.2. Base requirement. The principal objective of an authenticated key exchange protocol is to prevent the impersonation attack. Any such protocol does not guarantees that A and B will have a shared key when the protocol is completed. The protocol guarantees that A and B:

- share a common key
  `OR`
- share no key with each other OR anyone else.

In what follows we will define the basic notions of protocols, secure protocols and desirable protocol characteristics [6, 7, 9, 12]. Let $I = \{1, \ldots, N\}$ be the set of entities in a computer network and let's assume that an adversary is not included in this set (all the parties from set I are honest).

Definition 2.1. *A protocol is a pair $P = (\Pi, G)$ of probabilistic polynomial-time computable functions (polynomial-time in their first input):*

Π *specifies how (honest) players behave;*

G *is a generator of key pairs implemented in every part of computer network, where* Π *has as input:*

- 1^k *the security parameter*
- $i \in I$ *identity of sender*
- $j \in I$ *identity of intended recipient*
- $K_{i,j}$ *i's key pair together with j's public value*
- *tran a transcript of the protocol run so far (i.e. the ordered set of messages transmitted and received by i so far in this run of the protocol)*

$\Pi(1^k, i, j, K_{i,j}, tran)$ *outputs a triple* (m, δ, K) *where:*

- *m message which is to be sent from i to j*
- *δ is i's current decision*
- *K the session-key established by i and j*

Definition 2.2. *A particular run of a protocol $P = (\Pi, G)$ is an insecure run if any party involved in the run, say A, executes the protocol faithfully, accepts the identity of another party, and either of the following conditions holds:*

- *At the time that A accepts the other party's identity (before she sends or receives a subsequent message), the other party's record of the partial or full run does not match A's record*
- *The exchanged key accepted by A is known to someone other than the party whose identity A accepted (This conditions does not apply to authentication without key exchange)*

Definition 2.3. *A protocol $P = (\Pi, G)$ is a secure AK protocol if:*

- *At the time that A accepts the other party's identity (before she sends or receives a subsequent message), the other party's record of the partial or full run matches A's record*
- *It is computationally infeasible for the agreement key accepted by A to be recovered by anyone other than A and possibly the party whose identity A accepted (This condition does not apply to authentication without key exchange)*

In accordance with [2, 6, 9, 20, 26] it is necessary for the protocol fulfill to fulfill requirements:

- *Known-key security* - in order to protect the communication confidentiality, every run of the protocol should create a new and secret key K, named *session key.* A protocol should still achieve its goal in the face of an adversary who has learned some other session key
- *Unknown key-share resilience* - A cannot be obliged into sharing a key with 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-compromise impersonation resilience* - Suppose A's long-term private key is disclosed and an adversary E knows this value. In this case E can now impersonate A, since it is precisely this value that identifies A. However, it may be desirable in some circumstances that this loss does not enable E to impersonate other entities to A
- *Forward-secrecy* - Even if the long-term private key of A is disclosed, the secrecy of previous session key established by honest entities is not affected
- *Key control* - neither A nor B can enforce the other to accept a predefined key or a to predetermine any portion of that

With the purpose of producing a protocol which can be above the standards of the other already existing protocols it is necessary to decrease the number of calculations required and/or increase the number of the calculations necessary to crack the key. There were created numerous protocols based on the elliptic curves, most of them with a big necessity of computer calculations.

3. Secure Agreement Key ($S_{ECC}AK$)

3.1. Preliminaries. In this section we describe our ($S_{ECC}AK$) - proposed protocol. It is based on one public key and one private key as well as a system of elliptic curves for each party. The authentication technique used here is based on public key encryption and the establishment of the common key will be made with a random generated number (at every run session of protocol, for every party) and an elliptic curves system Let $E(Z_p)$, as in [24] be the domain parameter of the protocol, where E is an elliptic curve over a finite field Z_p and p is a large prime number. The number of rational points on an elliptic curve is finite and will be denoted by $\sharp E(Z_p)$, it should be divisible by a large prime n [8]. Let $P, Q \in E(Z_p)$ be two points, of order n, on the elliptic curve $E(Z_p)$. The selection of P and Q must be made with respect to the conditions imposed in [17], with $n > 2^{160}$ and $n > 4\sqrt{p}$. The number of rational points on the curve (i.e. $\sharp E(Z_p)$) should be under the relation (Hasse's Theorem): $(\sqrt{p} - 1)^2 \leq \sharp E(Z_p) \leq (\sqrt{p} - 1)^2$. The precaution needed in choosing the elliptic curves system is that n must not divide $p^k - 1$ for all $1 \leq k \leq S$, where S is large enough so that it is computationally infeasible to find discrete logarithms in Z_{p^S}, [4]. We also define a pair of keys S_i (secret key) and P_i (public key) for each party of the computer network ($i \in I = \{1, \dots, N\}$). The system (S_i, P_i) is for a public key encryption scheme.

We denote by

- $E_i^S(m)$ - encryption of message m with i's secret key
- $E_i^P(m)$ - encryption of message m with i's public key
- $h(\bullet)$ - hash function $SHA - 1$
- $m_1|m_2$ -adjoining of messages m_1 and m_2

3.2. Proposed protocol. The proposed protocol is one of challenge-response type and has in its structure two communication steps, that assure all required conditions presented above for a secure protocol. The first step is from A to B, and after the identity verification B proceeds to the second step: generates a response, sends it to A and calculates the shared key. When A receives the message it makes the identity verification for received message and, in case of success, it proceeds to the computation of the shared key.Public parameters will be: $(P_i, E(Z_p), P, Q, n)$ with $P, Q \in E(Z_p)$

Public functions:$E_i^S(m)$, $E_i^P(m)$, $h(\bullet)$

Private values for i:

- S_i - secret key
- d_A - generated value for each session key

Protocols:

– A make

a) generates a pseudo random number $d_A \in [1, n-1]$

b) calculates $A_1 = d_A(P^{-1} + Q) = (x^A_{1,y^A_1})$. Let $x = x_1 \; mod \; n$. If $x = 0$ then *goto a)*

c) calculates $A_2 = h(P_A|A_1)$

d) calculates $A_3 = E_A^S(A_2)$

First communication step (from A to B): A send to B $(A_1|A_3)$

– B make

i. calculates $s_1 = h(P_A|A_1)$

ii. calculates $s_2 = E_A^P(A_3)$. If $s_1 \neq s_2$ terminates the protocol run with failure.

iii. Generates pseudo random number $d_B \in [1, n-1]$

iv. calculates $B_1 = d_B(P^{-1} + Q) = (x_1^B, y_1^B)$. If $x = 0$ go to iii.

v. calculates $B_2 = h(P_B, B_1)$

vi. calculates $B_3 = E_B^S(B_2)$

vii. $K_B = d_B A_1 = (x_2^B, y_2^B)$

viii. $x = x_2^B$ mod n. If $x = 0$ then go to iii.

Second communication step (from B to A) B sends to A: $(B_1|B_3)$

– A make

e) calculates

$s_1 = h(P_B, B_1)$

$s_2 = E_B^P(B_3)$

f) If $s_1 \neq s_2$ terminates the protocol run with failure

g) $K_A = d_A B_1$

At the end of the protocol, if properly conditioned (without any active attack from an adversary)

$K_A = K_B = K$ is the basic point in order to construct the common key for secure communication between A and B (with symmetrical encryption).

4. Security Consideration

In the case which is treated in this paper, we assume like in [14] that all parties of the network are regular participants and there is an adversary which has access to all network communications. A regular party, which means an honest one, does not publish its secret value and/or does not publish in the network its generated values. In accordance with the definitions presented in [15, 16] the attacks an adversary can mount may be classified as follows:

> a) *passive attack* - this kind of action refers only to the actions which an adversary makes in order to intercept all messages over the network communication system without any change of the original messages
> b) *active attack* - in this case an adversary intercepts the communications but he can also makes modification in the original messages and/or inject (in the network) other messages

Proposition 4.1. *In protocol* $(S_{ECC}AK)$ *it is computationally infeasible for the key accepted by A, to be recovered by another party except B.*

Proof. In accordance to the above model of elliptic curves that was described we have all the circumstances exposed in [8]: given an elliptic curve E defined over Z_p, a point $P \in E(Z_p)$ and a point $H \in E_{Z_P}$, we need to establish an integer ν, $\nu \leq 0 \leq n-1$, such that $H = \nu P$. Hence that it is computationally infeasible to determine the integer ν. It is easy to see that the recovery of the key means to recover the integer ν. □

Proposition 4.2. *It is computationally infeasible to impersonate the authenticity of the parties which participate at protocol* $(S_{ECC}AK)$.

Proof. From the protocol description we can observe that the problem of impersonating a party it is the same with the well-known problem of the discrete logarithm. □

Theorem 4.1. *The protocol* $(S_{ECC}AK)$ *is a secure protocol.*

Proof. The validity of the theorem results from Proposition 4.1 and Proposition 4.2. □

In [1, 10, 13] it is shown how it is possible to break another kind of system, but with the some level of security like our proposal protocol. The necessary time is $\simeq 2^{21}$ years.
In [23] it is illustrate an parallel implementation of Pollard rho-method, where 1000 processors are used in an implementation in $Z_{2^{155}}$, and their conclusion is that 1500 years are necessary to find all points.

5. Conclusions

In this paper we propose an elliptic curve based protocol for an authenticated key agreement and we discuss its security. It is a well-known fact that the secrecy of a key based on an elliptic curves system depends on power computation that an adversary can dispose, and security of the proposal protocol in this paper is constructed with respect to this fact. What we done in this paper is a new protocol which increase the number of steps needed by the adversary in order to find the session secret key and to increase the difficulty of impersonation.

References

[1] G. Agnew, R. Mullin, S. Vanstone, An implementation of elliptic curve cryptosystem over $F_{2^{155}}$, IEEE Journal on Selected Areas in Communications, Vol. 11 (1993), pp. 804-813.

[2] William Aiello, Steven M. Bellovin, Matt Blaze, Ran Canetti, John Ioannidis, Angelos D. Keromytis, Omer Reingold, *Efficient, DoS-Resistant, Secure Key Exchange for Internet Protocols* , Proc. ACM Computer and Communications Security (CCS) Conference. November 2002, Washington, DC. (pp 48-58).

[3] ANSI X9.42,*Agreement of Symmetric Algorithm Keys Using Diffie-Hellman* , 2001.

[4] ANSI X9.62,*The Elliptic Curve Digital Signature Algorithm (ECDSA)* , 1999.

[5] ANSI X9.63,*Elliptic Curve Key Agreement and Key Transport Protocols* , 2001.

[6] Mihir Bellare, Lenore Cowen, Shafi Goldwasser,*On the Structure of Secret Key Exchange Protocol* , Advances in Cryptology - Crypto '89, Springer - Verlag 1990, LNCS 435, pp. 604 - 605.

[7] Mihir Bellare, Phillip Rogaway,*Entity authentication and Key Distribution* , Advances in Cryptology - Crypto '93, Springer - Verlag 1994, LNCS 773, pp. 232 - 249.

[8] Ion Blake, Gadiel Seroussi, Nigel Smart,*Elliptic curves in cryptography* , London Mathematical Society, LNS 265, pg 34-42, 79-87.

[9] Simon Blake-Willson, Don Johnson, A. Menezes,*Key Agreement Protocols and their Security Analysis* , The Sixth IMA International Conference of Cryptography and Coding, Cirencester, England, 17-19 December 1997, 1355 (1997), 30-45.

[10] Certicom White Paper,*The elliptic curve cryptosystem for smart card* , Published: May 1998.

[11] W. Diffie, M.E. Hellman,*New directions in cryptography* , IEEE Transactions on Information Theory 22 (1976), 644-654.

[12] Whitfield Diffie, Paul C. van Oorschot, Michael J. Wiener,*Authentication and Authenticated Key Exchanges* , Kluwer Academic Publishers, Designs, Codes and Cryptography, Vol. 2 (1992), 107-125.

[13] Shuhong Gao, Joachim Von Zur Gathen, Daniel Panario, Victor Shoup,*Algorithms for Exponentiation in Finite Fields* , Journal of Symbolic Computation (2000) 29, 879-889.

[14] Joshua D. Guttman, F. Javier Thayer, Lenore D. Zuck,*The Faithfulness of Abstract Protocol Analysis: Message Authentication* , Journal of Computer Security, 2003.

[15] Jonathan C. Herzog,*The Diffie-Hellman Key-Agreement Scheme in the Strand-Space Model* (This work supported by the National Security Agency), Proceedings of 16th IEEE Computer Security Foundations Workshop, June 2003.

[16] Jonathan C. Herzog,*Computational soundness of formal adversaries* , Master's thesis, Massachusetts Institute of Technology, 2002.

[17] F. Hess, G. Seroussi, N. P.,*Two topics in hyperelliptic cryptography* , Smart. Technical Report CSTR-00-008, Department of Computer Science, University of Bristol, June 2000.

[18] Chang Hyi, Jong-In Lim, Jeong-Soo Kim, IEEE P1363.*Contribution to Project develops Standard Specifications For Public-Key Cryptography, An Efficient and Secure Key Agreement.*

[19] Anna M. Johnson, Peter S. Gemmell,*Authenticated Key Exchange Provably Secure against the Man-in-the-Middle Attack,* Proceedings of the sixth IMA International Conference on Cryptography and Coding, Lecture Notes in Computer Science, Springer Verlag, 1355 (1997), 30-45, Journal of Cryptology (2002) 15: 139-148.

[20] Laurie Law, Alfred Menezes, Minghua Qu, Jerry Solinas, Scott Vanstone,*An Efficient Protocol for Authenticated Key Agreement Designs* , Kluwer Academic Publishers, Codes and Cryptography, Vol. 28 (2003), 119-134.

[21] C. Lim, P. Lee,*A key recovery attack on discrete log-based schemes using a prime order subgroup* , Advances in Cryptography 1997,Springer-Verlag, Lecture Notes in Computer Science, Vol. 1294 (1997), pp. 275-288.

[22] U. Maurer,*Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms* , Advances in Cryptology - Crypto '94, Springer-Verlag (1994), pp. 271-281.

[23] Paul C. van Oorschot, Michael J. Wiener,*Parallel Collision Search with Cryptanalytic Applications* , Springer - Verlag, ournal of Cryptology (1999) Vol. 12, pp. 1-28.

[24] Michael Rosing,*Implementing Elliptic Curve Cryptography* , MP Co, 1998, 103-126.

[25] Shahrokh Saeednia, Rei Safavi-Naini,*Efficient Identity-Based Conference Key Distribution Protocols* , ACISP'98, Springer- Verlag, 1998, LNCS 1438, pp. 320-331.

[26] Jin Wook Byun, Ik Rae Jeong, Dong Hoon Lee, Chang-Seop Park,*Password-Authenticated Key Exchange between Clients with Different Passwords* , R. Deng et al. (Eds.), Springer-Berlin Heidelberg, ICICS 2002, LNCS 2513, pp. 134-146, 2002.

[27] Y. Yacobi, *A key distribution paradox* , Advances in Cryptology 1990, Lecture Notes in Computer Science, Springer-Verlag 1991, Vol. 537, pp. 268-273.

(Nicolae Constantinescu) Department of Informatics, University of Craiova,
Al. I. Cuza street, 13, Craiova RO-200585, Romania, Tel/Fax: 40-251412673
E-mail address: nikyc@central.ucv.ro