

A SECURE KEY AGREEMENT PROTOCOL USING ELLIPTIC CURVES

C. Popescu*

Department of Mathematics, University of Oradea
Str. Armatei Romane 5, Oradea, Romania
E-mail: cpopescu@uoradea.ro

Abstract

In this paper we propose a secure protocol for authenticated key agreement based on Diffie-Hellman key agreement, which works in an elliptic curve group. We also present a simpler authenticated key agreement protocol than the proposed one and a multiple key agreement protocol which enables the participants to share two or more keys in one execution of the protocol. We prove that our protocols meet the security attributes under the assumption that the elliptic curve discrete logarithm problem is secure.

Key Words: Protocols, authenticated key agreement, elliptic curves.

1. Introduction

In a key agreement protocol two or more distributed entities need to share some key in secret, called session key. This secret key can then be used to create a confidential communication channel amongst the entities. Numerous Diffie-Hellman-based key agreement protocols have been proposed over the years [1], [2], [3], [4], [5]. But many of them have turned out to be flawed [6], [7]. A number of desirable attributes of key agreement protocols have also been identified [8] and nowadays most protocols are analyzed with such attributes. These are the same ones used in [2], [3]:

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

*The author is presently at "Centre for Quantifiable Quality of Service in Communication Systems" (Q2S), NTNU, Trondheim, Norway. The centre is appointed Centre of Excellence by The Research Council of Norway. It is financed by the Research Council, NTNU and UNINETT, and supported by Telenor.

- **(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 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.** Neither entity should be able to force the session key to a preselected value.

The authors in [2], [3] proposed a new authenticated key agreement protocol which has some computational advantage with about 2.5 integer multiplications for each entity. However, Kaliski showed in [9] that this protocol does not possess the unknown key-share attribute. We note that pairings on elliptic curves have recently been used to create several key agreement protocols [10], [11], [12]. The Smart's protocol [10] requires a trusted key generation centre and it has the novel property that the key generation centre is able to recover the agreed session keys from the message flows and its secret key. Shim [13] pointed out that the Smart's protocol does not provide the forward secrecy attribute. Also, Joux [14] presented a tripartite key agreement protocol which makes use of pairings on elliptic curves and which requires each entity to make only a single broadcast. The advantage of the Joux's tripartite protocol over any previous tripartite key agreement protocol is that a session key can be established in just one round. The disadvantage is that the Joux's protocol is unauthenticated and suffers from man-in-the-middle attacks [15], [16]. Al-Riyami and Paterson [15] presented several protocols to provide authenticity to tripartite key agreement. Their proposals belong to the MTI [17] and MQV [2], [3] family of protocols. The main idea is to use certificates of the three entities, which are issued by a Certificate Authority, to bind an entity's identity with his static long-term key. Then the final session key is generated by both ephemeral keys and static keys. The authenticity of the static keys assures that only the entities who possess the static keys are able to compute the session keys. However, Shim [18] showed that the Al-Riyami and Paterson's protocols are vulnerable to the man-in-the middle attack, key compromise impersonation attack and several known-key attacks. Recently, Nalla and Reddy proposed in [19] three one-round tripartite ID-based key agreement protocols using the ideas of both ID-based approach and the key agreement protocol of Joux. Shim [20] showed that the Nalla-Reddy's protocol are insecure against man-in-the-middle attacks.

In this paper, we propose a secure key agreement protocol based on Diffie-Hellman key agreement [21], and we prove that it has the desirable attributes discussed in [8]. We also present a simpler authenticated key agreement protocol than the proposed one and a multiple key agreement protocol which enables the participants to share two or more keys in one execution of the protocol. The protocols presented in this paper have been described in the setting of the group of points on an elliptic curve defined over a finite field. Suitable choices include the multiplicative group of a finite field, subgroups Z_n^* , where n is a composite integer, and subgroups of Z_q^* of prime order q . Elliptic curve groups are advantageous because they offer equivalent security as the other groups but with smaller key sizes and faster computation times.

2. Our Key Agreement Protocols

Many researchers have examined elliptic curve cryptosystems, which were firstly proposed by Miller [22] and Koblitz [23]. The elliptic curve cryptosystems which are based on the elliptic curve discrete logarithm problem over a finite field have some advantages over other systems: the key size can be much smaller than those in the other schemes since only exponential-time attacks have been known so far if the curve is carefully chosen [24], and the elliptic curve discrete logarithms might be still intractable even if factoring and the multiplicative group discrete logarithm turn out to be tractable problems. The elliptic curve discrete logarithm problem is defined as follows.

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

In this section we describe the proposed authenticated key agreement protocol (AKAP) which is specified by the key generation and the protocol description. Also, we present a simpler authenticated key agreement protocol (SAKAP) than the proposed one.

2.1 Key Generation

In this paper we use an elliptic curve E defined over a finite field F_q of characteristic p . Firstly, we choose elliptic curve domain parameters (see [25]):

1. a field size q , where q is a prime power (in practice, either $q = p$, an odd prime, or $q = 2^m$).
2. two field elements $a, b \in F_q$, which define the equation of the elliptic curve E over F_q (i.e., $y^2 = x^3 + ax + b$ in the case $p > 3$, where $4a^3 + 27b^2 \neq 0$).
3. two field elements x_p and y_p in F_q , which define a finite point $P = (x_p, y_p)$ of prime order in $E(F_q)$ ($P \neq O$, where O denotes the point at infinity).
4. the order n of the point P .

The elliptic curve domain parameters can be verified to meet the following requirements [2], [3]. In order to avoid the Pollard-rho [26] and Pohling-Hellman [27] algorithms for the elliptic curve discrete logarithm problem, it is necessary that the number of F_q -rational points on E , denoted $\#E(F_q)$, be divisible by a sufficiently large prime n . To avoid the reduction algorithms of Menezes, Okamoto and Vanstone [28] and Frey and Ruck [29], the curve should be non-supersingular (i.e., p should not divide $(q + 1 - \#E(F_q))$). To avoid the attack of Semaev [30] on F_q -anomalous curves, the curve should not be F_q -anomalous (i.e., $\#E(F_q) \neq q$).

The operation of the key generation is as follows:

1. Chooses a one-way hash function, H , such as SHA-1 [31].
2. Selects random integers s_A, s_B from the interval $[1, n - 1]$. The value s_A is a secret key of the user A and s_B is the secret key of the user B .
3. Computes the points $Y_A = -s_A \cdot P$ and $Y_B = -s_B \cdot P$, which are the public key of a user A and B respectively.
4. Selects ID_A and ID_B , which are the identity information of a user A and B respectively.

2.2 Protocol Description

The authenticated key agreement protocol (AKAP) between A and B is as follows:

1. A generates random integers r_A, k_A (ephemeral keys) from the interval $[1, n - 1]$ and computes Q_A, V_A , points on E , such that:

$$Q_A = r_A \cdot P, \quad V_A = -k_A \cdot P \quad (1)$$

A sends the point V_A to B .

2. B randomly selects integers r_B, k_B (ephemeral keys) from the interval $[1, n - 1]$ and computes Q_B, V_B , points on E , such that:

$$Q_B = r_B \cdot P, \quad V_B = -k_B \cdot P \quad (2)$$

B computes $e_B = H(x_{Q_B}, x_{V_B}, x_{V_A}, ID_B, ID_A)$ and $d_B = r_B + e_B k_B + e_B s_B$, where x_{Q_B} is the x -coordinate of Q_B , x_{V_A} is the x -coordinate of V_A and x_{V_B} is the x -coordinate of V_B . B sends V_B, e_B, d_B to A .

3. A computes the point U_B , such that $U_B = d_B \cdot P + e_B \cdot (V_B + Y_B)$ and checks if $e_B = H(x_{U_B}, x_{V_B}, x_{V_A}, ID_B, ID_A)$. If it does not hold, then A terminates the execution. Otherwise, A computes:

$$e_A = H(x_{Q_A}, x_{V_A}, x_{V_B}, ID_A, ID_B) \quad (3)$$

$$d_A = r_A + e_A k_A + e_A s_A \quad (4)$$

where x_{U_B} is the x -coordinate of U_B , x_{V_B} is the x -coordinate of V_B , x_{Q_A} is the x -coordinate of Q_A and x_{V_A} is the x -coordinate of V_A . A computes the point K_A , such that:

$$K_A = -k_A \cdot V_B \quad (5)$$

and sends e_A, d_A to B .

4. B computes the point U_A , such that $U_A = d_A \cdot P + e_A \cdot (V_A + Y_A)$ and checks if $e_A = H(x_{U_A}, x_{V_A}, x_{V_B}, ID_A, ID_B)$. If it does not hold, then B terminates the execution. Otherwise, B computes:

$$K_B = -k_B \cdot V_A \quad (6)$$

The shared secret is the point $K = K_A = K_B$.

Note: Based on the assumption that the public keys Y_A and Y_B are always authentic and can not be forged, it is possible to design a simpler authenticated key agreement protocol (SAKAP) than the proposed one as follows. Let $K_S = -s_B \cdot Y_A = -s_A \cdot Y_B = s_A s_B \cdot P$ be the long term secret key shared by A and B . A generates a random integer k_A from the interval $[1, n - 1]$, computes $V_A = -k_A \cdot P$, $e_A = H(x_{V_A}, x_{K_S})$ and sends V_A and e_A to B . Similarly, B randomly selects an integer k_B from the interval $[1, n - 1]$, computes $V_B = -k_B \cdot P$, $e_B = H(x_{V_B}, x_{K_S})$ and sends V_B, e_B to A . As a result, A and B achieve the same shared secret key $K = -k_A \cdot V_B = -k_B \cdot V_A = k_A k_B \cdot P$ and also authenticate each other.

3. The Multiple Key Agreement Protocol

In this section we present a multiple key agreement protocol which enables the participants to share two or more keys in one execution of the protocol. The key generation is the same as in Section 2. The multiple key agreement protocol between A and B is as follows.

1. A generates random integers $r_A, k_{A_1}, \dots, k_{A_n}$ from the interval $[1, n - 1]$ and computes the points $Q_A, V_{A_i}, i = 1, \dots, n$, such that:

$$Q_A = r_A \cdot P, \quad V_{A_i} = -k_{A_i} \cdot P \quad (7)$$

A sends the points $V_{A_i}, i = 1, \dots, n$ to B .

2. B randomly selects integers $r_B, k_{B_1}, \dots, k_{B_n}$ from the interval $[1, n - 1]$ and computes $Q_B, V_{B_i}, i = 1, \dots, n$ such that:

$$Q_B = r_B \cdot P, \quad V_{B_i} = -k_{B_i} \cdot P \quad (8)$$

B computes:

$$e_B = H(x_{Q_B}, x_{V_{B_1}}, \dots, x_{V_{B_n}}, x_{V_{A_1}}, \dots, x_{V_{A_n}}, ID_B, ID_A) \quad (9)$$

$$d_B = r_B + e_B \sum_{i=1}^n k_{B_i} + e_B s_B \quad (10)$$

where x_{Q_B} is the x -coordinate of Q_B , $x_{V_{B_i}}$ is the x -coordinate of V_{B_i} and $x_{V_{A_i}}$ is the x -coordinate of V_{A_i} , $i = 1, \dots, n$. B sends $V_{B_i}, i = 1, \dots, n, e_B, d_B$ to A .

3. A computes the point U_B , such that $U_B = d_B \cdot P + e_B \sum_{i=1}^n V_{B_i} + e_B \cdot Y_B$ and checks if $e_B = H(x_{U_B}, x_{V_{B_1}}, \dots, x_{V_{B_n}}, x_{V_{A_1}}, \dots, x_{V_{A_n}}, ID_B, ID_A)$. If it does not hold, then A terminates the execution. Otherwise, A computes:

$$e_A = H(x_{Q_A}, x_{V_{A_1}}, \dots, x_{V_{A_n}}, x_{V_{B_1}}, \dots, x_{V_{B_n}}, ID_A, ID_B) \quad (11)$$

$$d_A = r_A + e_A \sum_{i=1}^n k_{A_i} + e_A s_A \quad (12)$$

where x_{Q_A} is the x -coordinate of Q_A , $x_{V_{A_i}}$ is the x -coordinate of V_{A_i} , $x_{V_{B_i}}$ is the x -coordinate of V_{B_i} , $i = 1, \dots, n$. A computes the points K_{A_i} , such that:

$$K_{A_i} = -k_{A_i} \cdot V_{B_i}, \quad i = 1, \dots, n \quad (13)$$

and sends e_A, d_A to B .

4. B computes the point U_A , such that $U_A = d_A \cdot P + e_A \sum_{i=1}^n V_{A_i} + e_A \cdot Y_A$ and checks if $e_A = H(x_{U_A}, x_{V_{A_1}}, \dots, x_{V_{A_n}}, x_{V_{B_1}}, \dots, x_{V_{B_n}}, ID_A, ID_B)$, where x_{U_A} is the x -coordinate of U_A , $x_{V_{A_i}}$ is the x -coordinate of V_{A_i} , $x_{V_{B_i}}$ is the x -coordinate of V_{B_i} , $i = 1, \dots, n$. If it does not hold, then B terminates the execution. Otherwise, B computes:

$$K_{B_i} = -k_{B_i} \cdot V_{A_i}, \quad i = 1, \dots, n \quad (14)$$

The shared secret keys are the points $K_i = K_{A_i} = K_{B_i}$, $i = 1, \dots, n$.

The presented multiple key agreement protocol may find use in some future novel application.

4. Security of our Key Agreement Protocols

We prove that our protocols meet the desirable attributes discussed in [8] under the assumption that the elliptic curve discrete logarithm problem is secure.

Theorem 1. *Our key agreement protocols have the desirable attributes: known-key security, (perfect) forward secrecy, key-compromise impersonation, unknown key-share and key control.*

Proof. Known-Key Security: If the two entities A and B execute the regular protocol run, then they clearly share their unique session key K as above.

(Perfect) Forward Secrecy: During the computation of the session key K for each entity, the random integers r_A, k_A, r_B, k_B still act on it. An adversary who captured their private keys s_A or s_B would have to extract the random integers (ephemeral key) r_A, k_A, r_B, k_B from Q_A, V_A, Q_B, V_B to know the previous or next session key between them. But, this is the elliptic curve discrete logarithm problem.

Key-compromise Impersonation: Now, suppose that the long-term private key s_A of the user A is disclosed. An adversary who knows this value can clearly impersonate A . Also, the adversary impersonates B to A knowing B 's long-term private key s_B . For the success of the impersonation, the adversary must know A 's ephemeral keys r_A and k_A . Also, in this case, the adversary would have to extract r_A and k_A from A 's ephemeral public value, Q_A and V_A , to generate the same session key K with A . This also is the elliptic curve discrete logarithm problem.

Unknown Key-Share: Suppose an adversary C tries to make A believe that the session key is shared with B , while B believes that the session key is shared with C . To launch the unknown key-share attack, the adversary C would have to set his public key to be certified even though he does not know his correct private key. In this sense, the adversary C uses the public values (points) Y_A, Y_B and P . Let $f_t(R_1, \dots, R_l) = \sum_{i=1}^l t_i R_i$, where R_i 's are points on E and $t = (t_1, \dots, t_l)$ are integers from the interval $[1, n-1]$. Then C should set his public key Y_C as $Y_C = f_t(Y_A, Y_B, P)$. Suppose C got the value Y_C certified as his public key and suppose the following generalized model for unknown key-share attack. Suppose that $V_C = f_p(Y_A, Y_B, P, V_B)$ and $V'_C = f_m(Y_A, Y_B, P, V_A)$, where $p = (p_1, \dots, p_l)$ and $m = (m_1, \dots, m_l)$ are integers from the interval $[1, n-1]$. For C to launch the unknown key-share attack successfully, he should force A and B to share the same secret session key $K = K_A = K_B$ through the protocol run. In practice, through the protocol run, A and B get their session key K_A and K_B respectively as those in the following:

$$K_A = -k_A \cdot V_B, \quad K_B = -k_B \cdot V'_C \quad (15)$$

The adversary C does not know s_A, s_B, k_A, k_B even though C can control the integer values t_i, p_i, m_i . The adversary C can force the equation $K_A = K_B$ to hold for many values of k_A and k_B . Now we can consider the following equation as an identical one for the variables k_A and k_B :

$$k_A \cdot V_B = k_B \cdot V'_C \quad (16)$$

We can change this equation as the form $a \cdot P = O$, by unfolding the values V_A, Y_C, V_C, V'_C with respect to P . Then we can not solve this equation for t_i, p_i, m_i , since we do not have sufficient information on s_A, s_B, k_A, k_B .

Key Control: The key-control is impossible for the third party. The only possibility of key-control attack may be brought out by the participant of the protocol B . But, for the party B to make the party A generate the session key

K_B which is pre-selected value by B , for example B should solve the equation $K_B = -k_B \cdot V_A$. This is the elliptic curve discrete logarithm problem. \square

Further on, we compare our protocols with other protocols (see Table 1). The complexity is measured by the number of dominant computations (e.g. modular exponentiation or integer multiplication with a point in elliptic curve cryptography) for each party. We assume that in our protocols, the ephemeral keys of each party (and the points Q_A, V_A, Q_B, V_B) are generated in a precomputation phase.

In the following, we use some abbreviations for convenience such as:

- MQV-is Protocol 1 of Law, Menezes, Qu, Solinas and Vanstone [2], [3].
- ECDH-is elliptic curve version of Diffie-Hellman’s protocol [32], [33].
- MQVKC-is the MQV protocol with key confirmation [34].
- MTI/A0, MTI/C0-are protocols of Matsumoto, Takashima and Imai [17].
- Joux-is the protocol of Joux [14].
- Smart-is the protocol of Smart [10].
- UKA-Unknown Key-share Attack
- MMA-Man in the Middle Attack
- SSA-Small Subgroup Attack.
- FSA-Forward Secrecy Attack

Table 1: Complexity comparison for various protocols

Protocol	Complexity	Attacks
AKAP	3	none known
SAKAP	1	none known
MQVKC	2.5	none known
MQV	2.5	UKA
ECDH	2	MMA
MTI/A0	3	UKA
MTI/C0	2	SSA
Joux	2	UKA
Smart	4	FSA

The elliptic curve version of Diffie-Hellman’s protocol (ECDH) [32] requires 2 integer multiplications per entity and is insecure against man-in-the-middle attacks [33]. The MTI/A0 and MTI/C0 protocols require 3 and 2 integer multiplications, respectively. The MTI/A0 protocol is vulnerable to the unknown

key-share attack and the MTI/C0 is insecure against small subgroup attack [2], [3]. Also, the Joux's protocol requires 1 integer multiplication and 1 Weil pairing operation (evaluating the Weil pairing is a more costly operation than a half of a point multiplication). The Joux's protocol is unauthenticated and suffers from man-in-the-middle attacks [15], [16]. The Smart's protocol requires 2 integer multiplications and 2 Weil pairing operations. However, Shim showed in [13] that the Smart's protocol does not provide the forward secrecy attribute.

Blake-Wilson and Menezes [34] showed that the MQV protocol with key confirmation (MQVKC) is resistant to unknown key-share attacks. Our AKAP protocol provides all security attributes as well as the MQV protocol with key confirmation [34]. The MQV protocol with key confirmation requires 2.5 modular exponentiations per entity, while our protocol requires 3 integer multiplications. Because the MQV protocol with key confirmation requires in addition 1 message authentication code algorithm (MAC), we can deduce that our AKAP protocol is as efficient as MQVKC protocol. The AKAP protocol has been presented to provide the desirable security attributes which are not provided by the MQV, ECDH, MTI/A0, MTI/C0, Joux and Smart protocols. Also, the SAKAP protocol requires only 1 integer multiplication per entity. So, the SAKAP protocol (which can replace the AKAP protocol) is more efficient than the MQV protocol with key confirmation.

5. Conclusion

In this paper we proposed a secure protocol for authenticated key agreement AKAP based on Diffie-Hellman key agreement, which works in an elliptic curve group. A disadvantage is that each participant has to generate two random numbers from the interval $[1, n - 1]$ in one execution. Another disadvantage is that it requires entities slightly more modular exponentiations or integer multiplications than other protocols. Also, we presented a simpler authenticated key agreement protocol SAKAP than the proposed one which is more efficient than the MQV protocol with key confirmation. We proved that our protocols meet the security attributes under the assumption that the elliptic curve discrete logarithm problem is secure.

6. Acknowledgements

The author would like to thank prof. Svein J. Knapskog as well as the anonymous referees for their valuable comments.

References

- [1] M. Bellare, P. Rogaway, Entity Authentication and Key Distribution, *Proceedings of CRYPTO'93*, Santa Barbara, USA, 1994, 341-358.

- [2] L. Law, A. Menezes, M. Qu, J. Solinas, S. Vanstone, An efficient Protocol for Authenticated Key Agreement, *Technical Report CORR98-05*, Department of CO, University of Waterloo, 1998.
- [3] L. Law, A. Menezes, M. Qu, J. Solinas, & S. Vanstone, An efficient Protocol for Authenticated Key Agreement, *Designs, Codes and Cryptography*, 28(2), 2003, 119-134.
- [4] A. Menezes, M. Qu, S. Vanstone, Key Agreement and the need for authentication, *Proceedings of PKC'95*, Toronto, Canada, 1995.
- [5] A. Menezes, M. Qu, S. Vanstone, Some new key agreement protocols providing mutual implicit authentication, *Proceedings Workshop on Selected Areas in Cryptography (SAC'95)*, Nashville, USA, 1995, 22-32.
- [6] M. Burmester, On the risk of opening distributed keys, *Proceedings of CRYPTO'94*, Santa Barbara, USA, 1994, 308-317.
- [7] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography* (Boca Raton, FL: CRC press, 1997).
- [8] S. Blake-Wilson, D. Johnson, A. Menezes, Key Agreement Protocols and Their Security Analysis, *Proceedings of Sixth IMA International Conference on Cryptography and Coding*, Cirencester, UK, 1997, 30-45.
- [9] B. Kaliski, An unknown key-share attack on the MQV key agreement protocol, *ACM Transaction on Information and Systems Security*, 4(3), 2001, 275-288.
- [10] N.P. Smart, An identity based authenticated key agreement protocol based on the Weil pairing, *Electronic Letters*, 38(13), 2002, 630-632.
- [11] F. Zhang, S. Liu, K. Kwangjo, ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings, *Proceedings of IEEE International Symposium on Information Theory*, Yokohama, Japan, 2003, 136-148.
- [12] C. Popescu, I. Mang, An Authenticated Key Agreement Protocol Based on the Weil Pairing, *Proceedings of International Conference on Applied Informatics (AI 2003)*, Innsbruck, Austria, 2003, 797-800.
- [13] K. Shim, Efficient ID-based authenticated key agreement protocol based on Weil pairing, *Electronic Letters*, 39(8), 2003, 653-654.
- [14] A. Joux, A one-round protocol for tripartite Diffie-Hellman, *Proceedings of Algorithmic Number Theory Symposium*, Leiden, The Netherlands, 2000, 385-394.
- [15] S.S. Al-Riyami, K.G. Paterson, Tripartite authenticated key agreement protocols from pairings, *Proceedings of IMA Conference on Cryptography and Coding*, Cirencester, UK, 2003.

- [16] K. Shim, Efficient one round tripartite authenticated key agreement protocol from Weil pairing, *Electronic Letters*, 39(2), 2003, 208-209.
- [17] T. Matsumoto, Y. Takashima, & H. Imai, On seeking smart public-key distribution systems, *The Transactions of the IECE of Japan*, 69(2), 1986, 99-106.
- [18] K. Shim, Cryptanalysis of Al-Riyami-Paterson's Authenticated Three Party Key Agreement Protocols, *Cryptology ePrint Archive, Report 2003/122*, available at <http://eprint.iacr.org>, 2003.
- [19] D. Nalla, K.C. Reddy, ID-based tripartite Authenticated Key Agreement Protocols from pairings, *Cryptology ePrint Archive, Report 2003/004*, available at <http://eprint.iacr.org>, 2003.
- [20] K. Shim, Cryptanalysis of ID-based Tripartite Authenticated Key Agreement Protocols, *Cryptology ePrint Archive, Report 2003/115*, available at <http://eprint.iacr.org>, 2003.
- [21] W. Diffie, & M. Hellman, New directions in cryptography, *IEEE Trans. Inform. Theory*, 22(6), 1976, 644-654.
- [22] V. Miller, Uses of elliptic curves in cryptography, *Proceedings of Crypto'85*, Santa Barbara, USA, 1986, 417-426.
- [23] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, 48, 1987, 203-209.
- [24] N. Koblitz, CM-Curves with Good Cryptographic Properties, *Proceedings of Crypto'91*, Santa Barbara, USA, 1992.
- [25] C. Popescu, An identification scheme based on the elliptic curve discrete logarithm problem, *Proceedings of The Fourth International Conference/Exhibition on High Performance Computing Asia-Pacific Region*, Beijing, China, 2000, 624-625.
- [26] J. Pollard, Monte Carlo methods for index computation *mod p*, *Mathematics of Computation*, 32, 1978, 918-924.
- [27] S. Pohling, & M. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, *IEEE Transactions on Information Theory*, 24, 1978, 106-110.
- [28] A. Menezes, T. Okamoto, & S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Transactions on Information Theory*, 39, 1993, 1639-1646.
- [29] G. Frey, & H. Ruck, A remark concerning m -divisibility and the discrete logarithm in the divisor class group of curves, *Mathematics of Computation*, 62(206), 1994, 865-874.

- [30] I. Semaev, Evaluation of discrete logarithms in a group of p -torsion points of an elliptic curve in characteristic p , *Mathematics of Computation*, 67, 1998, 353-356.
- [31] National Institute of Standards and Technology, Secure Hash Standard (SHS), *FIPS Publication 180-1*, 1995.
- [32] M. Aydos, E. Savas, C.K. Koc, Implementing network security protocols based on elliptic curve cryptography, *Proceedings of the Fourth Symposium on Computer Networks*, Istanbul, Turkey, 1999, 130-139.
- [33] W. Mao, *Modern Cryptography: Theory and Practice* (Englewood Cliffs, NJ: Prentice-Hall, 2003)
- [34] S. Blake-Wilson, A. Menezes, Authenticated Diffie-Hellman Key Agreement Protocols, *Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC '98)*, Kingston, Canada, 1999, 339-361.

Constantin Popescu was born at Danesti, Romania, on 21st October, 1967. He received the MSc. degree in Computer Science from the University of Timisoara, Timisoara, Romania, in 1992. In 1992 he became an Assistant Professor at the Department of Mathematics, University of Oradea, Oradea, Romania. Since 1998 he has been a Lecturer at the Department of Mathematics, University of Oradea. In 2001 he has received the Ph. D degree in Computer Science (cryptography) at the Babes-Bolyai University, Cluj Napoca. Since 2003 he has been an Associate Professor at the Department of Mathematics, University of Oradea. His current research interests include cryptography, network security, security protocols, group signatures, identification schemes.