RES MILITARIS
REVUE EUROPEENNE D ETUDES EUROPEAN JOURNAL OF MILITARY STUDIES
Social Science Journal

Tripartite Key Agreement Protocol Using Decomposition Search Problem Authenticated by Digital Signature

By

T.Isaiyarasi
Assistant Professor Department of Mathematics, SRM Valliammai Engineering College, SRM Nagar, Kattankulathur Chengalpat District,Tamil Nadu India
Email: isaiyarasit.maths@srmvalliammai.ac.in

S.Chitra
Associate Professor Department of Mathematics, SRM Valliammai Engineering College, SRM Nagar, Kattankulathur Chengalpat District,Tamil Nadu India

N.Sundarakannan
Assistant Professor Department of Mathematics, SRM Valliammai Engineering College, SRM Nagar, Kattankulathur Chengalpat District,Tamil Nadu India

V.Vijayalakshmi
Assistant Professor Department of Mathematics, SRM Valliammai Engineering College, SRM Nagar, Kattankulathur Chengalpat District,Tamil Nadu India

Abstract

Key Agreement protocol (KAP) is one of the fundamental cryptographic primitives after encryption and digital signature which enables two or more entities arrive at a common key, which may be later used for any cryptographic purpose. In this paper we proposed an authenticated tripartite key agreement protocol. The decomposition search problem plays the role of one way function. The five dimensional Discrete Heisenberg group is chosen as the platform group which is non abelian. The KAP proposed in this paper is authenticated by digital signature. Thus, the proposed KAP satisfies all the necessary security attributes required.

Keywords — Key agreement protocol, Decomposition search problem, Digital signature**.**

Introduction

The intrinsic difficulty of key establishment in large computer networks has led to the invention of Public key cryptography where one of the keys can be made public. There are two well-known categories of key establishment protocols; namely the key transport and key agreement. Key transport enables two communicating parties to obtain a common secret key by using pre- established secure communication channels between them and a trusted third party. Key agreement is preferred to key transport as in Key agreement all the communicating parties contribute information to arrive at a shared secret key. Thus KAP s are of central interest in security world. In this paper an authenticated tripartite KAP is proposed in which the Factorization search problem is chosen as the one way function which is from combinatorial group theory and to use the search problems as one way function a non abelian group plays the role of a platform group and in this paper the five dimensional discrete Heisenberg group is used.

The paper is organized as follows section 2 introduces the discrete Heisenberg group (DHG) and gives the computational facts, In section 3 the Decomposition search problem (DSP) and its intractability is discussed. In Section 4 a digital signature algorithm using DSP is presented and in section 5 a tripartite KAP authenticated by Digital signature is proposed, section 6 briefs the security analysis and section 7 concludes the paper.

The Five Dimensional Discrete Heisenberg Group

$\mathcal{H} = Z_p^5$ be the set of all elements of the form $(x_1, x_2, x_3, x_4, x_5)$ and the binary operation in this set is defined as follows.

$$(x_1, x_2, x_3, x_4, x_5) \cdot (y_1, y_2, y_3, y, y_5)$$
$$= (x_1 + y_1, x_2 + y_2, x_3 + y_3, x_4 + y_4, x_5 + y_5 + x_1 y_3 + x_2 y_4$$

This operation satisfies the closure, associative properties and under this operation an identity element exists and each element possesses an inverse. But this operation is not commutative, thus this set constitutes a non abelian group which is infinite.

This group is made finite as follows:

$$(x_1, x_2, x_3, x_4, x_5) \cdot (y_1, y_2, y_3, y, y_5)$$
$$= (x_1 + y_1, x_2 + y_2, x_3 + y_3, x_4 + y_4, x_5 + y_5 + x_1 y_3 + x_2 y_4) mod\ p$$

where p is a prime number.

Computational Facts:

(i) Order of $\mathcal{H} = Z_p^5$ is P^5

(ii) Identity element is $e = (0,0,0,0,0)$

(iii)Inverse:$(x_1, x_2, x_3, x_4, x_5)^{-1} = (-x_1, -x_2, -x_3, -x_4, -x_5 + x_1 x_3 + x_2 x_4) mod p$

(iv) $(x_1, x_2, x_3, x_4, x_5)^n =$

$$\left(nx_1, nx_2, nx_3, nx_4, nx_5 + n^{(2)}(x_1 x_3 + x_2 x_4)\right) mod p$$

(v)Generators of $\mathcal{H}$ are $(1,0,0,0,0), (0,1,0,0,0), (0,0,1,0,0), (0,0,0,1,0)(vi)[a, b] = a \cdot b \cdot a^{-1} \cdot b^{-1} = (0,0,0,0, x_1 y_3 + x_2 y_4 - x_3 y_1 - x_4 y_2) mod\ p$(vii)Thus any element of $\mathcal{H}$ is expressed as follows: $(x_1, x_2, x_3, x_4, x_5) = (1,0,0,0,0)^{x_1} \cdot (0,1,0,0,0)^{x_2} \cdot (0,0,1,0,0)^{x_3} \cdot (0,0,0,1,0)^{x_4} \cdot [(1,0,0,0,0), (0,0,1,0,0)]^{x_5}$(viii) Order of any element is p, that is for any $a, a^p = e(identity)$

(ix)Subgroups of DHG: The cyclic subgroup $G_1 =< e, a >$ is of order P, the cyclic subgroup $G_2 =< e, a, b >$ is of order $P^2$where $a \cdot b = b \cdot a$, the cyclic subgroup $G_3 =< e, a, b, c >$ where a, b, c commute with each other is of order P^3, … the cyclic subgroup $H_{p-1} =< e, a_1, a_2, \dots, a_{p-1} >$ where the generators commute with each other is of order P^{P-1}

III. DECOMPOSITION SEARCH PROBLEM (DSP):

Given a non- abelian group G and two subgroups $A, B \leq G$ and two elements $x, y \in G$ find any two elements $a \in A, b \in B$ that would satisfy $a \cdot x. b =$ y provided atleast one such pair exist.

DSP In Discrete Heisenberg Group:

Given $x\ and\ y$ such that $y = a \cdot x \cdot b,\ \ x \in A, y \in B$ the DSP is to find a and b

Intractability Of DSP:

Let $a = (a_1, a_2, a_3, a_4, a_5)$, $b = (b_1, b_2, b_3, b_4, b_5)$, $x = (x_1, x_2, x_3, x_4, x_5)$, $y = (y_1, y_2, y_3, y_4, y_5)$

and $y = a \cdot x \cdot b$

$$(y_1, y_2, y_3, y_4, y_5) = (a_1, a_2, a_3, a_4, a_5) \cdot (x_1, x_2, x_3, x_4, x_5) \cdot (b_1, b_2, b_3, b_4, b_5)$$

$$= (a_1 + x_1 + b_1,\ a_2 + x_2 + b_2, a_3 + x_3 + b_3, a_4 + x_4 + b_4, a_5 + x_5 + b_5 + a_1x_3 + a_2x_4 + a_1b_3 + x_1b_3 + a_2b_4 + x_2b_4) \bmod p$$

$$y_1 = a_1 + x_1 + b_1, y_2 = a_2 + x_2 + b_2\ , y_3 = a_3 + x_3 + b_3\ , y_4 = a_4 + x_4 + b_4\ ,$$

$$y_5 = a_5 + x_5 + b_5 + a_1x_3 + a_2x_4 + a_1b_3 + x_1b_3 + a_2b_4 + x_2b_4$$

To find a and b one needs to solve the above set of equations.

The possible ways to get x_1, y_1 are listed as follows

If $y_1 - x_1 = p - 1$

a_1	$p-1$	0	$p-2$	1	…	$(p-1)/2$
b_1	0	$p-1$	1	$p-2$	…	$(p-1)/2$

There are p such possibilities available

If $y_2 - x_2 = p - 2$

a_1	$p-2$	0	…	$(p-3)/2$	$(p-1)/2$
b_1	0	$p-2$	…	$(p-1)/2$	$(p-3)/2$

There are $p - 1$ such possibilities available.

Proceeding in similar way

If $y_1 - x_1 = 1$

a_1	0	1
b_1	1	0

There are 2 possibilities

If $y_1 - x_1 = 0$

a_1	0	1	$p-1$	…	$(p+1)/2$	$(p-1)/2$
b_1	0	$p-1$	1	…	$(p-1)/2$	$(p+1)/2$

There are p possibilities available

Similar arguments arise in the case of the other 3pairs of elements namely,(a_2, b_2) (a_3, b_3) (a_4, b_4) and but when one tries to get (a_5, b_5) from

$y_5 = a_5 + x_5 + b_5 + a_1x_3 + a_2x_4 + a_1b_3 + x_1b_3 + a_2b_4 + x_2b_4$ it is much more complicated.

Thus to get the secret values x and y from one needs to search all the elements of the group when p is chosen large it may become hard to achieve.

Iv. Digital Signature Algorithm (Dsa)

Digital signatures enable the recipient of the information to verify the authenticity of the information's origin, and also verify that the information is intact. Thus, digital signatures provide authentication and data integrity. A digital signature also provides non-repudiation, which means that it prevents the sender from claiming that he or she did not actually send the information. These features are every bit as fundamental to cryptography as privacy, if not more. A digital signature serves the same purpose as a handwritten signature. However, a handwritten signature is easy to counterfeit. It consists of three algorithms namely Key Generation, Signature Generation and Signature Verification Algorithms

INITIAL SET UP:
Communicating Parties: A, B
Platform group: Discrete Heisenberg Group
One way function: Decomposition search problem
Public information: Group element $x_1 \in G$ and cyclic subgroups $G_1 =<$ e, $g_1, g_2, g_3 >$ and $G_2 =<$ e, $h_1, h_2, h_3 >$ where $g_i g_j = g_j g_i$ and $h_i h_j = h_j h_i$ for i = 1,2,3 and j = 1,2,3
Cryptographic hash function: h
DIGITAL SIGNATURE ALGORITHM BETWEEN A AND B
Key Generation:
A chooses $\alpha_1 \in G_1, \beta_1 \in G_2$ and computes $S_1 = \alpha_1 x_1 \beta_1$and made it as public B chooses $\alpha_2 \in G_1, \beta_2 \in G_2$ and computes $S_2 = \alpha_2 x_1 \beta_2$ and made it as publicSignature Generation:
1. A computes $S_Æ = a_1 S_2 \beta_1$
2. B computes $S_B = \alpha_2 S_1 \beta_2$
3. A chooses $\alpha_3 \in G_1, \beta_3 \in G_2$, computes $S_3 = \alpha_3 S_B \beta_3$ and sets S_3 as public and (α_3, β_3)as private key pair
4. B chooses $\alpha_4 \in G_1, \beta_4 \in G_2$, computes $S_4 = \alpha_4 S_Æ \beta_4$, and sets S_4 as public and (α_4, β_4) as private key pair
5. A computes $K_Æ = \alpha_3 S_4 \beta_3$
6. Let 'm_1' be the message to be sent by A to B and 'h' be a cryptographic hash functionAcomputes $y_1 = h\,(m_1)$
7. A signs the message m_1 by computing $S_5 = S_Æ y_1 K_Æ$and sends $(S_5 ,_1)$ to B

Verification Algorithm:
On receiving (s_5, m_1), B computes $S_B^{-1} S_5 K_B^{-1} = y_1$, also he computes $h(m_1)$, if $h(m_1) = y_1$ he accepts the message, otherwise he rejects the message
Since $S_A = S_B$ and $K_A = K_B$ are the common keys Bob gets $S_B^{-1} K_A K_B^{-1} = S_B^{-1} S_A Y_1 K_A K_B^{-1} = y_1$
Digital signature algorithm between B and C:
Key generation
B chooses $\alpha_5 \in G_1, \beta_5 \in G_2 and\ x_2 \in G$ and computes $R_1 = \alpha_5 x_2 \beta_5$
C chooses $\alpha_6 \in G_1, \beta_6 \in G_2$ and computes $R_2 = \alpha_6 x_2 \beta_6$
Signature generation
B computes $R_B = \alpha_5 R_2 \beta_5$, C computes $R_C = \alpha_6 R_1 \beta_6$
B chooses $\alpha_7 \in G_1, \beta_7 \in G_2$ and computes $R_3 = \alpha_7 R_C \beta_7$ and made it publicC chooses $\alpha_8 \in G_1, \beta_8 \in G_2$ and computes $R_4 = \alpha_8 R_B \beta_8$ and made it publicA computes $L_B = \alpha_7 R_4 \beta_7$ and B computes $L_C = \alpha_8 R_3 \beta_8$.
$L_B = L_C$ is their common key

Let m_2 be the message to be communicated with C, B computes $y_2 = h(m_2)$ and signs it by computing
$R_5 = R_B y_2$. B sends (R_5, m_2)
Verification:
On receiving (R_5, m_2) C verifies as follows.
C computes $T_C^{-1} T_B M_C^{-1}$ and he getsy_3, also he computes $h(m_3)$ and accepts $m_3$if $h(m_3) = y_3$
Correctness:
Since $T_A = T_C$ and $M_A = M_C$ are the common keys A gets $T_C^{-1} T_B M_C^{-1} = T_C^{-1} T_B y_3 M_A M_C^{-1} = y_3$
Digital Signature Algorithm for C and A
A chooses $\alpha_9 \in G_1, \beta_9 \in G_2 and\ x_3 \in G$ and computes
$T_1 = \alpha_9 x_3 \beta_9$
C chooses $\alpha_{10} \in G_1, \beta_{10} \in G_2$ and computes $T_2 = \alpha_{10} x_3 \beta_{10}$
Signature generation:
A computes $T_Æ = \alpha_9 T_2 \beta_9$
C computes $T_C = \alpha_{10} R_1 \beta_{10}$
A chooses $\alpha_{11} \in G_1, \beta_{11} \in G_2$ and computes $T_3 = \alpha_{11} T_C \beta_{11}$ and made it public
C chooses $\alpha_{12} \in G_1, \beta_{12} \in G_2$ and computes $T_4 = \alpha_{12} T_A \beta_{12}$
A computes $M_A = \alpha_{12} T_4 \beta_{12}$ and computes $M_C = \alpha_{13} T_3 \beta_{13}$
$M_A = M_C$ is their common key
Let m_3 be the message to be communicated with A, C computes $y_3 = h(m_3)$ and signs it by computing
$T_5 = T_C y_2 M_C$. B sends (T_5, m_3)
Verification:
On receiving (T_5, m_3) , A verifies as follows.
C computes $T_A^{-1} T_5 M_A^{-1}$ and he gets y_3, also he computes $h(m_3)$ and accepts m_3 if $h(m_3) = y_3$
Correctness:
Since $T_A = T_C$ and $M_A = M_C$ are the common keys A gets
$T_C^{-1} T_5 M_C^{-1} = T_C^{-1} T_B y_3 M_A M_C^{-1} = y_3$

V. Tripartite Key Agreement Protocol Authenticated By Digital Signature

Initial set up:

Communicating Parties: A, B and C

Platform group: Five dimensional Discrete Heisenberg group (DHG)One way function: Decomposition search Problem (DSP)

Public Information: group element z $\in G$ and cyclic subgroups $G_1 =<$ e, $g_1, g_2, g_3 >$ and

$G_2 =<$ e, $h_1, h_2, h_3 >$ where $g_i g_j = g_j g_i$ and $h_i h_j = h_j h$ for i = 1,2,3 and j = 1,2,3

Public Information: group element z $\in$ م and cyclic subgroups $G_1 =< e, g_1, g_2, g_3 >$ and

$G_2 =< e, h_1, h_2, h_3 >$ where $g_i g_j = g_j g_i$ and $h_i h_j = h_j h_i$ for $i = 1,2,3\ and\ j = 1,2,3$

Round I:
A chooses $a_1 \in G_1, b_1 \in G_2$ and $\in G_1$, computes $K_{11} = a_1 z b_1$ and $z_1 = h(K_{11})$

A signs his public key by using the signature as described in section 4, the signing keys for A (A to B) are

$S_A and\ K_A$ he signs K_{11} by computing $P_{11} = S_A z_1 K_A$, A sends (P_{11}, K_{11}) to B.

B chooses $a_2 \in G_1$, $b_2 \in G_2$, computes $K_{12} = a_2 z b_2$ and $z_2 = h(K_{12})$.B sings his public key by using the signature generated by him,(stated in section 4) , the signing keys for B (B to C) $S_Æ\ and\ K_Æ$, hesigns K_{12} by computing $P_{12} = R_B z_2 L_B$, A sends (P_{12}, K_{12}) to C.

C chooses $a_3 \in G_1$, $b_3 \in G_2$, computes $K_{13} = a_3 z b_3$ and

$z_3 = h(K_{13})$. C sings his public key by using the signature generated by him,(stated in section 4) , the signing keys for A (C to A) $T_C\ and\ M_C$, hesigns K_{13} by computing $P_{13} = T_C\ z_3 M_C$, A sends (P_{13}, K_{13}) to C.

After receiving the public keys from the other entities A, B and C verifies the keys received by using the verification algorithm given in section 4 as follows.

A verifies the public key from C by computing $T_A^{-1} P_{13} M_A^{-1} = T_A^{-1} T_C z_3 M_C M_A^{-1}$ also A computes $h(K_{13})$ and if it is equal to z_3 , he accepts the public key from C and proceeds for the further communications otherwise he rejects the public key and concludes that it is not from C , the procedure terminates.

B verifies the public key from A by computing $S_B^{-1} P_{11} K_B^{-1} = S_B^{-1} S_A z_1 K_A K_B^{-1} = z_1$ also he computes $h(K_{11})$ and if it is equal to z_1 , he accepts the public key from A and proceeds for the further communications, otherwise he rejects the public key and concludes that it is not from A , the procedure terminates.

C verifies the public key from B by computing $R_C^{-1} P_{12} L_C^{-1} = R_C^{-1} R_B z_2 L_B K_B^{-1} = z_2$ also he computes $h(K_{12})$ and if it is equal z_2, he accepts the public from A and proceeds for the further communications, otherwise he rejects the public key and concludes that it is not from B, the procedure terminates.

Round II : At the end of Round I the communicting parties have the following informaotion with them ; A posssesses $K_{11} = a_1 z b_1$, B Posseses $K_{12} = a_2 z b_2$, C possesses $K_{13} = a_3 z b_3$.

A computes $K_{21} = a_1 K_{13} b_1$ and $u_1 = h(K_{21})$ signs kt by computing $P_{21} = S_A\ u_1 K_A$ and sends (P_{21}, u_1) to B.

B computes $K_{22} = a_2 K_{11} b_2$ and $u_2 = h(K_{22})$ signs it by computig $P_{22} = R_B u_2 L_B$ and sends (P_{22}, u_2) to C.

C computes $K_{23} = a_2 K_{11} b_2$ and $u_3 = h(K_{23})$ signs it by computig $P_{23} = T_C u_3 M_C$ and sends (P_{23}, u_3) to A.

After receiving the public keys the communicating parties verify them by using the verification algorithm as described in section 4 as follows:

A verifies the authenticity of K_{23} by computing $T_A^{-1} P_{23} M_A^{-1} = T_A^{-1} T_C u_3 M_C M_A^{-1} = u_2$ also he computes $h(K_{23})$ and if it is equal to u_3, he accepts the public key from C and proceeds for the further communications, otherwise he rejects the public key and concludes that it is not from C, the procedure terminates.

B verifies the public key from A by computing $S_B^{-1} P_{21} K_B^{-1} = S_B^{-1} S_A u_1 K_A K_B^{-1} = z_1$; also he computes $h(K_{21})$ and if it is equal to u_1, he accepts the public key from A and proceeds for further communications, otherwise he rejects the public key and concludes that it is not from A, the procedure terminates.

C verifies the public key from B , by computing $R_C^{-1} P_{22} L_C^{-1} = R_C^{-1} R_B u_2 L_B L_C^{-1} = u_2$, also he computes $h(K_{22})$ and if it is equal to u_2, he accepts the public key from A and proeeds for the further communications, otherwise he rejects the public key and concludes that it is not from B, the procedure terminates.

At the end of Round II , A posseses $K_{23} = a_3 K_{11} b_3$, B posseses $K_{21} = a_1 K_{13} b_1$ and Cpossesses $K_{22} = a_2 K_{11} b_2$.

They arrive at the common key by computing $K_A = a_1 K_{23} b_1, K_B = a_2 K_{11}\, b_2, K_C = a_3 K_{22} b_3$.

The shared secret key is $KEY = a_1 a_2 a_3 b_3 b_2 b_1$

Security Analysis

Security analysis for Digital signature algorithm:
Data Forgery:

Suppose an Eavesdropper E tries to send $(m_1)_f$ instead of $m_1$to B. Assume that E has sent$(m_1)_f$ to B. While verifying B finds that $h((m_1))_f \neq y_1$, thus he rejects the message. Thus the data forgery attack is not possible,thus he rejects the message. Thus the data forgery attack is not possible.3

Signature Repudiation:

Suppose A denies that he has not sent the message to B then B may hope that it may have come from an eavesdropper E. If the signature is not from A then B receives the signature as (S_3', m). While verifying B computes $S_B^{-1} S_3' K_B^{-1} \neq S_A y K_A K_B^{-1}$. Thus A cannot deny that the signature is not hi own. Thus signature repudiation is not possible.

Existential Forgery Attack:

Eavesdropper E tries to creat at least one message / signature pair. If E tries to sign a message m from A to B, he must know the signature of A.

Suppose E signed the message m wih his own signature, he computes $(S_3)_f = S_A' y K_A'$ and sends $\left((S_3)_f, m\right)$ as the signature to B. As A used the decompostion search problem to create his signature, it is hard to solve. E cannot get S_3. E cannot sign any message or get the signture. Thus Existental forgery attack is nnot possible. Thus the digital signature algorithm satisfies all the necessary security attirbutes.

Security analysis for Key agreement protocol:
Perfect forward secrecy:

A Protocol is said to have perfect forward secrecy if compromise of long- term keys does not compromise the past session keys.

This KAP provides the perfect forward secrecy, since even if the long term keys such asS_A, S_B, S_C are compromised the adversary cannot find their secret shared key in the past session. The secret shared key is arrived with the help of the private keys of each entity as they are choosing different secret keys for each communication. Therefore even if the long time keys are disclosed at any point of communication that will not affect the session keys. Since the common key K= $a_1 a_2 a_3 z b_3 b_2 b_1$ requires the session secret keys (a_1, b_1) of A , (a_2, b_2) of B and (a_3, b_3) of C. Thus this protocol gives the perfect secrecy.

Known Key secrecy:

Each run of a key agreement protocol between two entities A and B should produce a unique secret key; such keys are called session keys. A protocol should still achieve its goal in the face of an adversary who has learned some other session keys. This KAP provides the known key secrecy as each session key is unique for each value of the session secret keys. For example if the adversary came to know about the common key of the three entities A, B and C

, that is $K = a_1a_2a_3b_3b_2b_1$which involves the session secret keys (a_1, b_1) of A, (a_2, b_2) of B and (a_3, b_3) of C. For the next communication they are going to choose the secret keys different from these keys. Thus the other session keys cannot be found out by the adversary as it involves session secret keys of each entity.

Key compromise impersonation:

Suppose any one of the communicating parties say A's long term private key is disclosed he may impersonate A to other entities, since it is precisely the identity of A. In this KAP the key compromise impersonation clearly impossible as the communicating parties sign their public keys by using the digital signature algorithm established between them each round. An adversary cannot take part in the communication between A,B and C.

Unknown key share: An unknown key-share attack on an authenticated key agreement protocol is an attack whereby an entity A ends up believing it shares a key with another entity B and although this is in fact the case that B mistakenly believes the key is instead shared with an entity C ≠ A. This scenario is impossible in this KAP as the entities A, B and C are using the DSA to sign their public keys. If suppose an adversary pretends as A to B then he faces the difficulty of getting the signature of A and sign A's public key. On verification B finds that it does not come from A and rejects the public key. Thus this KAP is secure against the unknown key share attack.

Key Control: Neither entity should be able to force the session key to a preselected value. As the common shared key is arrived with the contribution of each entity, this case will never arise. Thus this KAP provides no key control attribute.

VI. Computational Facts:

7.1 Group multiplication:

$$(x_1, x_2, x_3, x_4, x_5) \cdot (y_1, y_2, y_3, y, y_5)$$
$$= (x_1 + y_1, x_2 + y_2, x_3 + y_3, x_4 + y_4, x_5 + y_5 + x_1y_3 + x_2y_4) mod\ p$$

One group multiplication requires 5 integer additions, 2 integer multiplications and one modular operation The number of group multiplications required for the entities are summarized below.

Entity	Round I			Round II			Total
	Key	Signature	Verification	Key	Signature	Verification	
A	2	2	2	2	2	2	12
B	2	2	2	2	2	2	12
C	2	2	2	2	2	2	12

7.2 Storage requirements:

Entity A : Private Keys(a_1, b_1) , Space requirement 20 bytes and 40 bytes in case of long integers Public keys:K_{11}, K_{21}, space required is 20 bytes and 40 bytes in case of long integers. Thus each entity requires 20 bytes and 40 bytes in case of long integers for private as well as public keys.

Viii. Conclusion

In this paper a tripartite key agreement protocol is proposed which satisfies all the security attributes of an authenticated key agreement protocol .The Key agreement protocol is made authenticated by applying the digital signature of the communicating parties. The Digital Signature algorithms between the communicating parties are pre-established and later they are used to authenticate the public keys of the parties. Thus the common shared key arrived by the parties become a secured one. After arriving at a common key the communicating parties may use it for any future cryptographic communications. The key agreement protocol proposed here may be used in network security and cloud computing where there is a need for common key for cloud client and server.

References

Alexei Myasnikov,Vladimir Shpilrain, Alexander Ushakov,'Group Based Cryptography', CRM, BirkhäuserVerlag, Basel, Boston, Berlin.

Alfred Menezes, Paul Van Oorschot, Scott Vanstone, 'Handbook of Applied Cryptography',CRCPress Series on Discrete Mathematics and Its Applications.

I. Anshel, M. Anshel, and D. Goldfeld, 'An algebraic method for public-key cryptography', Math. Res. Lett., 6 (1999), 287–291

C.Birget, S. Magliveras and M. Sramka, 'On public-key cryptosystems based on combinatorial group theory', Tatra Mountains Mathematical Publications, 33 (2006),137–148.

Whitfield Diffie and Martin E. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, 22(6), 644–654. Doi: 10.1109/TIT.1976. 1055638

Lincoln Advisor: Byung-Jay Kahng ∗ August 5, 2009, Laura Janssen University of Nebraska - Extensions of the Heisenberg Group

Giuseppe Ateniese, Michael Steiner, and Gene Tsudik, Member, IEEE 'New Multiparty Authentication services and Key Agreement Protocols' IEEE Journal of Selected Areas in Communications, 18(4), (April 2000).

I.N. Herstein, 'Topics in Algebra', Xerox College Park, Lexington, Massachusetts, 2nd edition.

T.Isaiyarasi & K.Sankarasubramanian, A New Multiparty Key Agreement Protocol using triple decomposition problem in Discrete Heisenberg Group, International Journal of Computer Engineering and Information Technology, ISSN Online:0976-6375, Vol.4, Issue 6, 2013.

T.Isaiyarasi & K.Sankarasubramanian, Tripartite Key Agreement Protocol using Triple Decomposition Search Problem, International Journal on Cryptography and Information Security, ISSN: 1839-8626, Vol.2, No.1, 49-55, Sept 2012.

T.Isaiyarasi & K.Sankarasubramanian, A New Key Agreement Protocol using two layers of Security, Journal of Discrete Mathematical Sciences and Cryptography, Vol.15, No.2 & 3, April & June 2012.

T.Isaiyarasi & K.Sankarasubramanian, A New Multiparty Key Agreement Protocol using search problems in Discrete Heisenberg Group, Indian Journal of Computer Science and Engineering, ISSN: 2231-3850, Vol.3, Issue 1, 154-161. March 2012.

T.Isaiyarasi & K.Sankarasubramanian, A New Multiparty Key Agreement Protocol using search problems in Discrete Heisenberg Group, Indian Journal of Computer Science and Engineering, ISSN: 2231-3850, Vol.3, Issue 1, 154-161. March 2012.

A. Joux, 'A One Round Protocol for tripartite Diffe-Hellman', In W.Bosma,editor proceedings of Algorithmic Number Theory, Symposium, ANTS IV, Lecture Notes in Computer Science, 1838 (2000), 385−394,Springer Verlag

PeterJ.Khan, 'Automorphisms of the Discrete Heisenberg Group', arXiv:math / 0405109VI[math SG]6 (May2004).