A New Satellite Multicast Security Protocol Based on Elliptic Curve Signatures

Attila Altay Yavuz[1] , Fatih Alagöz[1] , Emin Anarım[2]
Computer Engineering Department, Bogazici University, 34342-Istanbul, Turkey[1].
({attila.yavuz, alagoz}@boun.edu.tr)
Electrical Engineering Department, Bogazici University, 80815-Istanbul, Turkey[2]
(anarim@boun.edu.tr)

Abstract

In this paper, we propose a new satellite multicast security protocol based on ECPVSS (Elliptic Curve Pintsov-Vanstone Signature Scheme). Our protocol is especially designed for satellite multicast systems having very large member size as well as highly dynamic member join-leave characteristic. We design two independent key distribution layered architecture that has many advantages over classical satellite multicast systems. Our protocol significantly reduces rekeying workload of satellite multicast systems. The number of keys that are stored on the satellite is also reduced. As a novel approach for secure key transmission, we utilize ECPVSS to provide major cryptographic goals while significantly reducing bandwidth consumption. We show that the proposed protocol can handle very large multicast system securely and effectively while providing additional advantages when compared to some widely accepted protocols.

1. Introduction

Providing security in satellite multicast systems is one of the most challenging problems in wireless communication. The problem becomes much severe for satellite multicast systems having very large number of members and high member join-leave frequency. Many different solutions have been proposed for the multicast security problem [1].Unfortunately, existing security models cause massive workload on all of the system components. To minimize the workload resulting from the security concern, we propose a novel multicast security protocol for use in satellite networks.

Our protocol is especially designed for satellite multicast systems having very large number of members and high member join-leave frequency. Our protocol consists of two new approaches and uses combined methods including new concepts for protocol designs and application suit cryptographic methods. Our protocol targets main source of the hierarchical key distribution protocol that is spreading the effect of the modification, which is performed on the single point of the logical key tree, to the whole tree. This problem stems from the fact that, in order to provide forward and backward security, for each member join leave event, group key must be updated in multicast security system.

Our protocol uses two independent key distribution layers for solving this frequently rekeying problem. First layer consists of satellite-terrestrial units (TUs) and second layer consists of TU-members. Both layer uses LKH (Logical Key Hierarchy) [2] key distribution protocol. Using two independent key distribution layers, effect of the modification is encapsulated on only its local group. Using independency principle, whenever a member join-leave event occurs for a member, only related terrestrial unit group is affected from that event. Any other part of the system is prevented from being modified that provides significant performance gain especially for satellite. Also, batch keying is done that decreases rekeying workload of the satellite.

Apart from protocol based performance gains, our protocol uses appropriate cryptographic algorithms that make two independent encryption layers feasible and secure. In this protocol, we use ECPVSS (Elliptic Curve Pintsov-Vanstone Signature Scheme) [3], [4] that satisfies many properties of ECDSA [5] like authentication, integrity and unforgeability. However, ECPVSS is a message recovery type signature that is especially suitable for bandwidth constraint environment [3] and has advantages compared to classical signature schemes. Our protocol uses ECPVSS to transmit session keys that will be used for batch keying and group key transmission. ECPVSS provides significant bandwidth usage advantages for satellite while providing high security. As far as our concern, ECPVSS has not been used for this purpose before.

2. Related Works

Key management protocols use hierarchical methods to handle large multicast systems. Essentially, key management protocols can be classified into two major categorizes [6]: Key based hierarchy and group based hierarchy. Key based hierarchy approaches use logical tree-structures for managing cryptographic keys in hierarchical manner such as [7], [8]. Group based approaches divide main group into hierarchical sub-groups in order to manage large multicast systems like [9], [10]. Integrating these approaches, hybrid methods exist such as [11].Our protocol is also a hybrid approach.

We compare our protocol with Flat and pure LKH protocol in section 6. Thus, we give some properties of these protocols. In Flat protocol, each member is directly connected to key manager and has a unique key.

0-7803-9521-2/06/$20.00 ©2006 IEEE.

Whenever a key update occurs, group key is sent to each member one by one encrypting it by unique keys of each member. Thus, key update and storage cost of Flat protocol is N. N *is the number of members in multicast* system. LKH protocol, designed for handling moderately large and dynamic groups, uses a tree structure to reduce rekeying cost. In LKH protocol, each member stores a key vector to reach group key. This key vector contains the keys which take place on the path of the member to reach the root of the tree. Using this method, for each member join-leave event, only keys that are on the affected paths are updated. This structure reduces rekeying cost of LKH from N to $k \log_k N$ where k is the branching factor of the tree. This is a significant advantage compared to the Flat protocol.

For cryptographic methods, generally, DLP (Discrete Logarithm Problem) based DH (Diffie-Hellmann) [12], public key cryptography algorithms such as RSA-ElGamal [13] and ECDH, which is extension of the DH in EC, are used. However, these approaches do not provide critical cryptographic goals together, which are confidentiality, authentication, integrity and unforgeability [14]. Especially, group based DH [15] and ECDH approach are vulnerable "man-in-the-middle attack". Classical digital signatures such as DSA and ECDSA [5] provide these properties but cause bandwidth overheads. Note that, ECC based cryptographic methods have important advantages for both computational complexity and key storage and are preferred for wireless networks [16]. For bulk data multicast, symmetric cryptography is used. Block ciphers in appropriate mode such as AES, DES or stream ciphers can be used.

.

3. Properties and Description of ECPVSS

ECPVSS is a message recovery (MR) type signature scheme based on ECC (elliptic curve cryptography). ECPVSS has many advantages for short messages when compared to the signature scheme with appendix [17] and some other MR type signature schemes.

ECPVSS has been proposed in [3] and is especially used for Digital Post Marking (DPM) applications. ECPVSS provides confidentiality, authentication, integrity and unforgeability together in efficient manner generating smaller signature sizes than classical digital signature algorithms.

Formal security proofs for ECPVSS are given in [18] covering ROM (Random Oracle Model). Also, some concrete examples are given for size of the messages used in DPM applications. Moreover, analysis, proofs and techniques for MR type signatures and ECPVSS can be found in [19]. Note that, ECPVSS has been standardized by IEEE in [20]. This standard also includes details about KDF (Key Derivation Function) which is used to obtain symmetric key from different data types.

We give definition and notations for ECPVSS algorithm. Let G be a public point of order n in the group of points on elliptic curve $E(\mathbb{F}_q)$ over finite filed $\mathbb{F}_q$ and number of points on the curve P is divisible by n. Then following notations are used:

γ: A point on the curve and used as implicit certificate. I_s : Identity of the signer. H : Cryptographic hash function, $\|$ denotes concatenation operation. a : Private key of the signer and is calculated by using I_s and γ. $Q = a \cdot G$: Public key of the signer.

$Data = C \| V$ where C represents data element that requires confidentiality and can be recovered during the verification. V is plaintext part of the data.

In the description of ECPVSS, we directly use Q and does not show how it is generated from I_s, γ and some other parameters. We say Q is authentically obtained to refer these processes.

Steps of ECPVSS are given below:

Signature Generation:

1. Split data into two part : V *and* C.
2. Generate a random number k where $k < n$.
3. $R = k \cdot G$, R is a point on the curve.
4. Derive a symmetric key R' from R using key derivation function. $R' = KDF(R)$.
5. Transform the C using bijective transformation Tr parameterized by R'. This transformation destroys the algebraic structure of C. Tr may be a symmetric encryption algorithm such as AES , DES or simply XOR operation: $e = Tr_{R'}(C)$.

   Confidentiality of R is protected by intractability of ECDLP and randomness of the value k.
6. $d = H(e \| I_s \| V)$.
7. $s = a \cdot d + k \bmod n$.
8. Pair (s, e) is the signature pair used for verification. Pair (s, e) and plaintext data part V are sent to the verifier.

Signature Verification:

1. Public key of signer Q is authentically obtained by verifier.
2. $d = H(e \| I_s \| V)$.
3. $U = s \cdot G - d \cdot Q$. Use KDF if necessary.
4. $X = Tr_U^{-1}(e)$. Recover the confidentially protected part of the data.
5. Check redundancy of X and if X has required redundancy declare $X = C$ and accept the signature as a valid signature.

4. Design Properties and Principles of Our Protocol

4.1. Contribution for Architectural Design

Most important performance gain is obtained from architecture design. *Our protocol uses two independent key distribution layers that provide significant performance gain for especially rekeying workload of the satellite.* In classical multicast systems, whenever a member join-leave event occurs, whole multicast system is affected from modification and group manager (satellite in our case) realizes key update according to the policy of key management protocol (LKH in our case). Under these conditions, if group manager is directly responsible from members then each member-join leave event inevitably affects to the group manager. This situation creates significant performance deterioration in large multicast groups. Also, rekeying workload especially becomes problem for satellite multicast system having dynamic mobile members. In long term, even if a good key management protocol is used, overall performance of system is determined by number of rekeying operation and number of members that are affected by rekeying operation.

Taking into consideration these problems, we design two independent LKH layers for satellite multicast security systems. In first layer (satellite- TU), satellite manages a TU group using LKH key management protocol. As long as TUs are available, satellite does not realize rekeying operation. In second layer (TU-members), each TU has its own member group and manages them using LKH key management protocol. *Whenever a member join-leave event occurs, only related TU is affected from modification. LKH rule is applied for key update to the local TU group. Neither other TU groups nor satellite are affected from modification. This approach significantly reduces the workload of the satellite.* Detailed analysis of our protocol is given in section 6.

LKH protocol is selected to use in layers because LKH can handle moderately large and dynamic multicast groups successfully. Note that, in our protocol, each local group contains small number of members due to independency principle.

4.2 Contribution for Cryptographic Method Aspect to the Multicast Security Protocols

In multicast security protocols, cryptographic methods that are used to transmit keys have critical importance. Even if key management protocol and architectural design minimize rekeying workload, if appropriate cryptographic methods are not used, then the system is overloaded due to cryptographic processes. This situation especially must be taken into consideration for layered architecture like our protocol.

Classical key exchange methods are prevalently used but naïve implementation of these protocols causes security problems. Note that, to provide major cryptographic goals, digital signature type cryptographic solutions are required. In classical signature with appendix applications, message is also transmitted with its signature. If message size is small, the signature of the message causes 100% overhead and creates significant bandwidth consumption.

Taking into consideration these factors, we propose a novel approach for cryptographic method to use in satellite multicast security protocol. To realize group key update, only small symmetric keys, which are generally 128 or 256, are transmitted to the destination. *Major idea behind of our choice is that an efficient MR type signature is excellent candidate to transmit these symmetric keys. ECPVSS is one of the most efficient MR type signatures and naturally possess the advantages of ECC as mentioned in section 2-3. As far as our concern, ECPVSS has not been used to transmit key update processes in satellite multicast security protocol.* Details of the advantages of the using ECPVSS are given in section 6.

5. Details of Our Protocol

We give details of our protocol. Following notations are used:

ICI: Implicit Certificate Information. EC_Keygen: Elliptic curve key generator validating parameters. $CPRNG$: Cryptographically secure pseudo-random number generator. N: Number of members in satellite multicast system. l: Number of TUs in satellite multicast system. n_l: Average number of members that belong(s) to a local TU member group. $E_K - D_K$: Symmetric encryption-decryption function.

5.1 Satellite-TU Layer

Satellite is responsible for generating and distributing group keys to the TUs in hierarchical manner. Also, satellite realizes data multicast using transmitted group key to the TUs. Satellite generates group key GK to realize data multicast and batch keying for group keys of TUs. GK is signed and recovered by ECPVSS that provides major cryptographic goals for satellite-TU layer. GK is used to for symmetric encryption of group key vectors z_i. TUs use group key vectors z_i to realize data multicast to the members. Note that, z_i and data are multicasted using symmetric key encryption functions.

1. Satellite generates implicit certificates and public-private pairs for each TU and inserts required keys to the ICIs. ICIs are transmitted to the TUs.

$Q_i = EC\_KeyGen(\gamma_i, I_{s_i}, a_i)$, $\mathrm{ICI_i} \leftarrow (Q_i$ and other required paramters) where $1 \le \mathrm{i} \le l$.

2. Satellite generates group key GK. GK is inserted to the C part and signed with ECPVSS. ICIs are

inserted into V part. Signature pairs for each TU are generated and transmitted to the TUs.

$$GK = CPRNG(),\ C = GK,\ V_i = ICI_i,$$
$$(e_i, s_i) = ECPVSS\_Sing(V_i, C),$$
$$TU_i \leftarrow (V_i, e_i, s_i),\ 1 \le i \le l.$$

3. Satellite generates group key vectors z_i for $1 \le i \le l$. Each group key vector z_i is assigned to a TU. Elements of group key vector z_i are $z_{i,j}$ where $1 \le i \le l$ and $1 \le j \le n_s$. $z_{i,j}$ denotes j'th group key that is used by i'th TU. Each TUs use $z_{i,j}$ group key to realize data multicast to the members. $z_{i,j}$ provides batch keying. Note that, satellite may store only seed values of the group key vectors in order to reduce number of keys that are stored. $z_{i,j}$ group keys are encrypted using GK.

$$z_{i,j} = CPRNG(l, n_s),\ z'_{i,j} = E_{GK}(z_{i,j}),$$
$$TU_i \leftarrow z'_{i,j},\ 1 \le i \le l,\ 1 \le j \le n_s.$$

4. Satellite realize data multicast using GK.

$$M' = E_{GK}(M),\ TU_i \leftarrow M'.$$

5. Whenever a TU join-leave event occurs, satellites update group key GK using ECPVSS as in previous steps while obeying LKH update rules.

5.2 TU-Member Layer

In this layer, each TU has its own local member group having 2048 member or more. TUs decrypt multicast data and z_i using GK that is obtained from satellite using ECPVSS. Each TU has its own z_i vector that contains group key vectors $z_{i,j}$. Suppose that, i'th TU uses $z_{i,j}$ group key in current state. After that, for i'th TU, whenever a member join-leave event occurs, TU uses next group key such that $z_{i,j} \rightarrow z_{i,j+1}$. Satellite is informed for group key modification. $z_{i,j}$ are used to encrypt multicast data and provides batch keying. $z_{i,j}$ are signed with ECPVSS and transmitted to the related members.

1. $(V_i, e_i, s_i, M') \leftarrow Satelllite.$
2. $C_i = ECPVSS\_Unsign(V_i, e_i, s_i),\ GK = C_i$ is obtained authentically and confidentiality by each TU.
3. Each TU obtains group key from satellite that will be used data multicast to the members: $z_{i,j} = D_{GK}(z'_{i,j})$.
4. Each TU decrypts multicast data using GK which is obtained from satellite: $M = D_{GK}(M')$.
5. Each TU generates implicit certificates and public-private pairs for each member and inserts required keys to the ICIs. ICIs are transmitted to the members.

$Q'_i = EC\_KeyGen(\gamma'_i, I'_{s_i}, a'_i)$, $\text{ICI}'_i \leftarrow (Q'_i$ and other required paramters) where $1 \le i \le n_s$.

6. Each TU transmits group key $z_{i,j}$ to its local member group using ECPVSS. Also, multicast data is encrypted using group key $z_{i,j}$.

$$C'_i = z_{i,j},\ V'_i = ICI'_i,\ 1 \le i, j \le n_s.$$
$$(e'_i, s'_i) = ECPVSS\_Sing(V'_i, C'_i),$$
$$M'' = E_{z_{i,j}}(M),\ Member_i \leftarrow (V'_i, e'_i, s'_i, M'').$$

7. Each member obtains group key $z_{i,j}$ from their TU using ECPVSS. Using group key $z_{i,j}$, each member decrypts multicast data using group key.

$$(V'_i, e'_i, s'_i, M'') \leftarrow Member_i,$$
$$C'_i = ECPVSS\_Unsing(V'_i, e'_i, s'),\ z_{i,j} = C'_i.$$

are obtained by each member. Then $M = D_{z_{i,j}}(M'')$.

8. *Whenever a member join-leave event occurs, only group key of related local member group is updated such that* $z_{i,j} \rightarrow z_{i,j+1}$ *applying LKH rule. Neither satellite nor other TU local groups are affected from modification. Satellite is only informed for next group key is in use.*

6. Performance Comparison and Results

6.1 Advantages and Performance Comparison for Architecture and Design Aspects

Our protocol has significant advantages to some well-known protocols for scalability, fast rekeying, and security aspects. Note that, most resource limited component of the system is satellite. Thus, it is critical to reduce workload of this component. Using two independent layers, satellite nearly is not affected from rekeying requirements of member and this situation significantly reduces the rekeying and cryptographic workload of the satellite.

Table 1 shows performance comparison of our protocol to the Flat and LKH protocol for five major criteria. Most important criterion is rekeying workload of the satellite. This criterion also determines computational effort and bandwidth consumption of the satellite. Rekeying workload of the satellite is determined by N and number of rekeying in certain time period, that is r. For instance, very large multicast system having $N = 10^6$ member, in moderate time period, $r = 10^5$ rekeying is a reasonable assumption.

In section 2, rekeying cost of Flat and LKH protocol are given as N and $k \log_k N$ respectively. In both protocol, due to members are only managed by a centralized group manager (satellite in our case), each rekeying operation (member join-leave event) affects to the satellite. Thus, for r rekeying, total rekeying workload of the satellite becomes $N \cdot r$ and $(k \log_k N) \cdot r$ for Flat and LKH protocol respectively. *In our protocol, satellite is only responsible for rekeying TUs, not members directly. Thus, satellite is not affected from* r. *This significantly reduces rekeying workload of the satellite.* Also, unlike to members, TUs do not show dynamic join-leave characteristic and rekeying workload coming from TUs is neglible. In addition to this, *our protocol uses advantages of the batch keying that reduces rekeying workload* (we show this contribution with parameter m). Having l TU, rekeying workload of satellite in our protocol is $(k \log_k l)/m$. *For aforementioned values, rekeying workload of satellite is* 10^{11}, $4*10^6$ *for Flat and LKH respectively. However, in our protocol, rekeying workload of the satellite is* $(2 \log_2 500) \approx 20$ *that is much smaller than Flat and LKH protocols (*m *=1).* When batch keying parameter m is increased, performance also increases.

Rekeying workload for TU can be found in same manner. Each TU manages a local member group having $n_l = N/l \approx 2048$ or more members. Whenever a member join leave event occurs, TU applies LKH rules to its local group. Number of rekeying for single TU group is $r_l = r/l$. Then, rekeying workload of a TU is $(\log_k (n_l)) \cdot r_l$. *This workload (approximately 2000-2500) is easily handled by even low capacity TU.*

Number of keys which are stored in satellite is another important parameter. In both Flat and LKH protocol, unique keys are stored in satellite for each member and storage load is N. In our protocol, satellite only stores seed values and unique keys for each TU together with a general group key. *Thus, number of keys stored in satellite is* $2 \cdot l + 1$ *which is much smaller than Flat and LKH protocols.* Number of keys, which are stored in one TU, are group key vector z_i together with LKH keys: $n_l + \log_k (n_l)$. Number of keys that are stored in member for Flat and LKH is 1 and $\log_k N$ respectively. In our protocol, it is $\log_k (n_l) \approx 11$. Table 1 summarizes these results.

6.2 Advantages of Selected Cryptographic Methods

As a novel approach, we use a message recovery type algorithm ECPVSS in our satellite multicast security protocol. Main goal of cryptographic routines in multicast security protocols is the securely transmitting group and session keys to their destination. These keys are generally 128-256 bit symmetric encryption keys. *For this reason, a message recovery digital signature algorithm based on ECC is very good choice to transmit these keys.*

Table 2 shows comparison of ECPVSS to other well-known cryptographic methods that are also frequently used in satellite multicast systems. Possible message length of ECPVSS signature and message overheads are given together with their alternatives. In our case, we assume that 128 or 160 bit group or session keys are transmitted to use in a block (AES or variable length block cipher) or stream cipher (LFSR based).

We firstly compare ECPVSS with 1024 bit RSA signature with appendix. Total length of the message and signature are 256 byte. Also, DSA with 1024 bit modulus and common signature with appendix size are shown. Note that, in Elgamal encryption, ciphertext is doubled [21]. Thus, encrypted session key (ciphertext) and signature sizes are 256 and 50 byte respectively. For ECDSA, order of EC is accepted as 20 byte and signature size is 50 byte. Like DSA, ciphertext is doubled in ECC. Thus, encrypted session keys and signature sizes are 50-100 and 50 bytes respectively. For DH and ECDH, common moduli are same with DSA and ECDSA. However, due to both parties of the communication send messages, total transmitted message is doubled and is 256 byte. For DH, session keys are encrypted with Elgamal cryptosystem while in ECDH they are encrypted with ECC. Note that, also RSA and El-Gamal type algorithms could be compared in message recovery type algorithms. However, implementing these algorithms with standardized appendix schemes is common application. Details can be found in [3]. Appropriate certificate sizes are selected for compared algorithms.

ECPVSS provides authentication, integrity and unforgeability while pure implementation of RSA-Elgamal, EC, DH and ECDH does not provide these properties. With appropriate key bit length, confidentiality can be provided by all methods. In our protocol, we insert certificate information to the plaintext part V. Also, due to group key is a cryptographically generated random number, it includes sufficient redundancy. However, considering worst case, we may add 10 byte redundancy. Using these, overhead of ECPVSS is 40-60 byte providing $2^{-80} - 2^{-160}$ total break resistance security. *Note that ECPVSS is at least 3 times better then nearest competitive for bandwidth consumption.* Table 2 summaries these results.

7. Conclusion

In this paper, we propose a new satellite multicast security protocol. Our protocol utilizes two independent key distribution layers (satellite-TU and TU-Members layers) that significantly reduce the rekeying workload of the satellite multicast system Also, number of keys that are stored in the satellite is reduced. This architecture

encapsulates rekeying operations in local member groups in TU-Member layer and effect of the modification does not spread to the whole multicast system and especially satellite. This approach provides scalability and high performance for especially very large and dynamic multicast systems. In addition to this, we introduce using ECPVSS as major cryptographic method in satellite multicast systems. ECPVSS is a MR type signature scheme based ECC and specifically designed for bandwidth limited environments. We use ECPVSS for secure group key and seed transmission that provides at least 3 times bandwidth advantages for best competitive method. Moreover, ECPVSS provides major cryptographic goals together efficiently while many methods can only partially provide these properties. We show that the proposed satellite multicast protocol is very promising one especially for highly dynamic and very large sized multicast members.

Acknowledgements

This work is supported by the State Planning Organization of Turkey under "Next Generation Satellite Networks Project", and Boğaziçi University Research Affairs.

8. References

[1] M. P. Howard, S. Iyengar, Z. Sun, H. Cruischank. Dynamics of Key Management in Secure Satellite Multicast, IEEE Journal on Selected Areas in Communications, Vol. 22, No.3, Feb 2004.

[2] D. Wallner, E. Harder, and R. Agee, Key management for multicast: Issues and architectures, IETF, RFC2627, June 1999.

[3] L. A. Pintsov and S. A. Vanstone. Postal revenue collection in the digital age., Proceedings of Financial Cryptography, FC'00, number 1962 in LNCS, pages 105-120. Springer-Verlag, 2000.

[4] D. Naccache and J. Stern. Signing on a postcard, Proceedings of Financial Cryptography , FC'00, number 1962 in LNCS, pages 121-135. Springer-Verlag, 2000.

[5] D. Johnson, A. Menezes. The Elliptic curve digital signature algorithm (ECDSA)", February 24, 2000.

[6] S. Mishra. Key management in large group multicast. Technical Report CU-CS-940-02, Department of Computer Science, University of Colorado, Boulder, CO., 2002.

[7] D. Balenson *et al.* Key management for large dynamic groups: One-way function trees and amortized initialization, IETF Draft, work-in progress, draft-balenson-groupkeymgmt-oft-00.txt, February 1999.

[8] A. Perrig, D. Song, and J.D. Tygar. ELK, a new protocol for Efficient Large-group Key distribution. IEEE Security and Privacy Symposium May 2001.

[9] S. Mittra. Iolus: A framework for scalable secure multicasting.In Proceedings of the ACM SIGCOMM'97, September 1997.

[10] S. Setia, S. Koussih, and S. Jajodia. Kronos: A scalable group rekeying approach for secure multicast. In Proceedings of the IEEE Symposium on Research in Security and Privacy, May 2000.

[11] J. Huang and S. Mishra. Mykil: A Highly scalable and efficient key distribution protocol for large group multicast. In the IEEE 2003 Global Communications Conference (GLOBECOM 2003), San Francisco, CA (December 2003).

[12] New Directions in Cryptography, W. Diffie and M. E. Hellman, IEEE Trans. Information Theory, vol. IT-22, Nov. 1976, pp: 644 654.

[13] Standard specifications for public key cryptography. IEEE P1363/D13, November 1999.

[14] B. Schneier, *Applied Cryptography*. New York: Wiley, 1996.

[15] M.Steiner, G. Tsudik, M. Waidner, "DH Key Distribution Extended to Groups", Proc. 3rd ACM Symp. on Computer and Communications Security, New Delhi, Vol. 1, pp31-37, March 1996.

[16] Kristin Lauter. The Advantages of elliptic curve cryptography for wireless security. IEEE Wireless Communications, February 2004.

[17]A. Menezes, P. van Oorschot, and S. Vanstone. Handbook of Applied Cryptography, CRC press", 1996.

[18] D. R. L. Brown and D. B. Johnson. Formal security proofs for a signature scheme with partial message recovery. Proceedings of CT-RSA'01,number 2020 in LNSC, pages 126{142. Springer-Verlag, 2001.

[19] Louis Granboulan, "PECDSA How to build a DL-Based digital signature scheme with the best proven security", NESSIE, October 2002.

[20] IEEE P1363a/D2. Standart specifications for public key cryptography: Pintsov-Vanstone Signature with message recovery, January 10, 2000.

[21] Douglas R. Stinson, *Cryptography, Theory and Practice, Second Edition*, CRC Press, 2002.

For very large System l > 500, k=2, r > 10^5 and N > 10^6 , $n_l = N / l$					
	Rekeying Load over Satellite	# of Keys Stored in Satellite	Rekeying Load for TU	# of Keys Stored in TU	# of Keys Stored in Member
FLAT	$N \cdot r$	N	-	-	1
LKH	$(k \log_k N) \cdot r$	N	-	-	$\log_k N$
Our Protocol	$(k \log_k l) / m$	$2 \cdot l + 1$	$(\log_k (n_l)) \cdot r_l$	$n_l + \log_k (n_l)$	$\log_k (n_l)$

Table 1. Performance comparison of our protocol to Flat and LKH protocols

Byte		RSA - Sig.		ElGamal –DSA		EC – ECDSA		DH	ECDH	**ECPVSS**
Size of the Transmitted Data for Rekeying(BW)	Session Key	128		256		50-100		256	50-100	**Included in Signature**
	Signature	128		50		50		256	100	**20**
	Certificate	256		168		60				**20**
	Total	512		474		160		512	150	**40-60**
Authentication		no	yes	no	yes	no	yes	no	no	**yes**
Integrity		no	yes	no	yes	no	yes	no	no	**yes**
Unforgeability		no	yes	no	yes	no	yes	no	no	**yes**
Confidentiality		yes								

Table 2. Advantages and properties of ECPVSS against its widely used alternatives