NOVEL METHODS FOR SECURITY MECHANISMS AND KEY MANAGEMENT TECHNIQUES IN WIRELESS NETWORKS BASED ON SIGNCRYPTION AND HYBRID CRYPTOGRAPHY

by

Attila Altay Yavuz

B.S., Computer Engineering, Yıldız Technical University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in FBE Program for which the Thesis is Submitted
Boğaziçi University
2006

NOVEL METHODS FOR SECURITY MECHANISMS AND KEY MANAGEMENT TECHNIQUES IN WIRELESS NETWORKS BASED ON SIGNCRYPTION AND HYBRID CRYPTOGRAPHY

APPROVED BY:

Assoc. Prof. Fatih Alagöz (Thesis Supervisor)	.....
Prof. Emin Anarım (Thesis Co-supervisor)	.....
Assist. Prof. Burak Acar	.....
Prof. Ufuk Çağlayan	.....
Prof. Cem Ersoy	.....

DATE OF APPROVAL: 14.06.2006

ACKNOWLEDGEMENTS

I would like to express my honest thanks to my advisor Assoc. Prof. Fatih Alagöz and my co-advisor Emin Anarım for their continuous help, invaluable comments and advanced contributions all through out my studies. They always motivate me and dedicate their valuable time to help me to complete my studies. With their support and guidance, I determine my way for future academic career.

I wish to thank all of my friends and teachers for their support. My special thanks must go to Taylan Saldıray and Ömür Kartal for their patient reviews of my studies. Also, I thank to Assoc. Prof. İhsan Yılmaz for encouraging me in my studies.

Finally, I want to express my love and gratefulness to my family for their endless love, support, encouragement, patience and self-sacrifice. I especially want to thank my father Prof. İlhami Yavuz for his advanced support and guidance in my studies and my mother biologist Ayşe Yavuz for her love and care to me. I also thank my elder sister Dr. Lale Yavuz for her supports and encouragements.

This work is supported by The State Planning Organization under "The Next Generation Satellite Networks and Applications" project, No: DPT 2003-K120250.

ABSTRACT

NOVEL METHODS FOR SECURITY MECHANISMS AND KEY MANAGEMENT TECHNIQUES IN WIRELESS NETWORKS BASED ON SIGNCRYPTION AND HYBRID CRYPTOGRAPHY

Providing security in wireless communication networks is one of the most challenging problems in security systems. Broadcast nature of wireless networks make them more vulnerable to eavesdropping and active attacks when compared to terrestrial fixed networks. Also, wireless networks are resource limited especially for power and bandwidth possibilities, which makes harder to provide security in these systems. These problems become much severe for wireless networks having very large number of members and high member join-leave characteristic.

In this thesis, in order to address aforementioned problems, we propose seven novel studies each of them provides efficient solutions for these problems in wireless networks. We especially focus on providing security in satellite networks and military Mobile Ad-hoc NETworks (MANET). We bring novelties to wireless network security systems in three main points: Structural design, integrated key management techniques and novel cryptographic approaches that have not been used in Secure Satellite Multicast Systems (SSMS) and military MANETs. Our structural design principles, integrated with hybrid key management techniques, are based on " independency of tiers " principle. In this principle, modification in a tier does not affect all other tiers in the network system. Our hybrid key management techniques combine centralized logical key tree based key management techniques and decentralized key management techniques in an efficient manner. We specifically utilized appropriate cryptographic methods to our security mechanisms.

We propose Two-Tier Pintsov-Vanstone Signature Scheme (TTPVSS), which introduces our independency of tiers principle and a novel hybrid key management technique. These approaches significantly reduce rekeying workload of satellites and provide many advantages when compared to traditional methods. Also, as a novelty, TTPVSS uses Elliptic Curve Pintsov-Vanstone Signature Scheme (ECPVSS), which provides high security and advantages. Then, we propose a new three-tier satellite multicast security mechanism based on Elliptic Curve Menezes-Qu-Vanstone (ECMQV). This security mechanism additionally uses special properties of GEO, MEO and LEO satellites for better performance and security. ECMQV, different from classical key exchange and digital signature schemes, achieves major cryptographic goals and security against active attacks. Our another study, NAMEPS, N-tier sAtellite Multicast sEcurity Protocol (Mechanism) based on Signcryption schemes uses N-tiered structure and Efficient Large Key management protocol (ELK) based hybrid key management technique, which further reduces rekeying and cryptographic workload of satellites. As a novel approach, NAMEPS uses a multi-recipient signcryption scheme, which provides computational and storage advantages. Apart from SSMS, we propose HIMUTSIS, HIerarchical MUlti-Tier adaptive ad-hoc network security protocol based on SIgncryption type key exchange Schemes for military MANETs. In HIMUTSIS, we propose a novel multi-tier structure for military MANETs, which reduces threshold cryptography requirement and single point of failure problems. Also, as a novelty, HIMUTSIS uses a multi-level security system and signcryption based key exchange protocols that provides high security and performance together. In addition to these, we also studied on improving some existing cryptosystems. In this sense, we propose IMC (Improved Merkle Cryptosystem), which has significant security advantages over both MC (Merkle Cryptosystem) and VMC (Variant of Merkle Cryptosystem). Security of IMC is compatible with today's modern public key cryptosystems. Apart from these, we work on STAKE (Signcryption Type Authentic Key Establishment), which integrates signcryption based approaches with our IMC algorithm. As a result, in this thesis, we present our major studies for wireless network security and cryptography in an integrated manner providing many advantages when compared to the traditional approaches.

ÖZET

KABLOSUZ AĞLARDA GÜVENLİK MEKANİZMALARI VE ANAHTAR YÖNETİM TEKNİKLERİ İÇİN SIGNCRYPTION VE HİBRİD KRİPTOGRAFİ TEMELLİ YENİ YÖNTEMLER

Kablosuz iletişim ağlarında güvenliğin sağlanması, güvenlik sistemlerinin en zorlu problemlerinden biridir. Kablosuz ağların tümyayın doğası, onları yerel sabit ağlara nazaran gizli dinleme ve aktif saldırılara karşı daha korunmasız kılmaktadır. Ayrıca, kablosuz ağlar özellikle enerji ve bant genişliği olanakları bakımından sınırlıdır ve bu durum kablosuz ağlarda güvenliği temin etmeyi güçleştirmektedir. Bu problemler, özellikle üye sayısının çok fazla olduğu ve üyelerin sisteme giriş-çıkışlarının sık gerçekleştiği dinamik karakteristikteki kablosuz ağlarda büyümektedir.

Biz bu tezde, yukarıda belirttiğimiz problemleri hedef alarak, her biri kablosuz ağlarda değindiğimiz sorunlara etkin çözümler sağlayan yedi yeni çalışma öneriyoruz. Bu çalışmaların ana konusunu, uydu ağlarında ve askeri tasarız mobil ağlarda güvenliğin sağlanması oluşturmaktadır. Biz, bu çalışmalarımızda kablosuz ağ güvenlik sistemlerine üç temel noktada yenilik getirdik: Yapısal tasarım, bütünleşik anahtar yönetim teknikleri ve bildiğimiz kadarıyla daha önce güvenli uydu çoklu yayın sistemleri ve askeri tasarsız mobil ağlarda kullanılmamış yenilikçi kriptografik yaklaşımlar. Sistemlerimizde yer alan karma anahtar yönetim teknikleriyle birleştirilmiş yapısal tasarım prensipleri "katmanların bağımsızlığı" prensibi üzerine kuruludur. Bu prensibe göre, katmanlarından herhangi birinde gerçekleşen değişiklik, diğer katmanlara sirayet etmemelidir. Karma anahtar yönetim tekniklerimiz, mantıksal anahtar ağacı temelli merkezi anahtar yönetim teknikleri ile dağıtık anahtar yönetim tekniklerinin etkin bir kombinasyonundan oluşmaktadır. Güvenlik mekanizmalarımızda kullanılan kriptografik yöntemler önerdiğimiz güvenlik mekanizmalarına uygun olarak seçilmiştir.

İlk çalışma olarak, yeni karma anahtar yönetim tekniklerimizi ve katmanların bağımsızlığı prensibini ortaya atan İki Katmanlı Pintsov-Vanstone İmzasını(TTPVSS) öneriyoruz. Bu yaklaşımlar, uydu üzerindeki yeniden anahtarlama yükünü önemli ölçüde azaltmakta ve geleneksel yöntemlere nazaran önemli avantajlar sağlamaktadır. Ayrıca, yenilik olarak TTPVSS, avantajlar sağlayan Eliptik Eğri Pintsov-Vanstone İmza Şemasını (ECPVSS) kullanmaktadır. Bu çalışmayı müteakiben, yeni bir Elliptik Eğri Menezes-Qu-Vanstone (ECMQV) temelli üç katmanlı uydu güvenli çoklu yayın mekanizması öneriyoruz. Bu güvenlik mekanizması, daha yüksek bir güvenlik ve iyi bir performans sağlamak amacıyla, GEO, MEO ve LEO uydularının kendilerine özgü niteliklerinden faydalanmaktadır. ECMQV, diğer klasik kriptografik yöntemlerden farklı olarak, temel kriptografik amaçlara ulaşmakta ve aktif saldırılara karşı da güvenliği temin edebilmektedir. Diğer çalışmamız olan NAMEPS, signcryption temelli ve çok katmanlı uydu çoklu yayın güvenlik protokolü, uyduların üzerindeki yükü daha da azaltmak amacıyla, çok katmanlı bir mimariyi ve özel karma anahtar yönetimi tekniklerini bir arada kullanmaktadır. NAMEPS geleneksel kriptografik yöntemlere göre önemli avantajlar sağlayan, çoklu alıcılı signcryption şemasını kullanmaktadır. Güvenli uydu çoklu yayın sistemleri dışında, askeri mobil tasarsız ağlarda güvenliği sağlamak üzere, HIMUTSIS (Signcryption tipi anahtar değişim şema temelli hiyerarşik çok katmanlı adaptif tasarsız ağ güvenlik protokolü)'i öneriyoruz. HIMUTSIS, askeri iletişim ağlarındaki tek noktaya bağımlılık problemini ve eşik değeri kriptografi gereksinimi azaltacak yeni bir çok katmanlı yapı önermektedir. Ayrıca, yeni bir yaklaşım olarak HIMUTSIS, yüksek performans ve güvenlik sağlayan çok seviyeli bir güvenlik sistemi ve signcryption temelli anahtar değişim protokollerini kullanmaktadır. Bu tez çalışmasında ağ güvenliği mekanizmalarına ek olarak, mevcut bazı kriptosistemlerin iyileştirilmesi üzerinde de çalıştık. Bu bağlamda, orijinal Merkle kriptosistemi (MC) ve onun bir varyantı üzerinde önemli güvenlik avantajları bulunan Geliştirilmiş Merkle Kriptosistemini (IMC) öneriyoruz. Buna ek olarak, signcryption temelli yaklaşımlarla IMC algoritmasını birleştiren STAKE (Signcryption tipi kimlik onaylı anahtar oluşturma) protokolü üzerinde de çalışmaktayız. Sonuç olarak, bu tez çalışmasında, bizler kablosuz ağlarda güvenlik ve kriptografi konularında geleneksel yaklaşımlara nazaran önemli avantajlar sağlayan temel çalışmalarımızı bütünleşik bir şekilde sunuyoruz.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET vi
LIST OF FIGURES xii
LIST OF TABLES xiii
LIST OF SYMBOLS/ABBREVIATIONS xiv
1. INTRODUCTION 1
1.1. Secure Group Communication in Wireless Networks 1
1.2. Content of This Thesis 3
2. KEY MANAGEMENT TECHNIQUES USED IN OUR MECHANISMS . . 7
2.1. Principles of Key Management Protocols for Secure Group Communication 7
2.2. Analysis of Key Management Techniques Used in Our Mechanisms . . 9
2.2.1. Centralized Key Management Protocols 9
2.2.1.1. Simple Key Management Protocol 10
2.2.1.2. Logical Key Hierarchy (LKH) 11
2.2.1.3. One-Way Function Tree (OFT) 13
2.2.1.4. Efficient Large-Group Key distribution (ELK) 15
2.2.2. Decentralized Key Management Protocols 16
2.2.2.1. Iolus 16
3. CRYPTOGRAPHIC TECHNIQUES USED IN OUR SECURITY MECHANISMS 17
3.1. Shortened Digital Signature Schemes (SDSS) 19
3.2. Signcryption Schemes 21
3.3. Achieving Major Cryptographic Goals in Key Establishment Protocols 25
3.4. Signcryption Based Key Establishment Protocols Used in Our Mechanisms 26
4. A NEW SATELLITE MULTICAST SECURITY MECHANISM BASED ON ELLIPTIC CURVE SIGNATURES 33
4.1. Introduction to TTPVSS 33
4.2. Properties and Description of ECPVSS 35

4.3. Design Properties and Principles of TTPVSS 37
4.3.1. Contribution for Structural Design 38
4.3.2. Contribution for Cryptographic Method Aspect to the Multicast Security Mechanisms 39
4.4. Details of Our Security Mechanism 39
4.4.1. Satellite-TU Tier 40
4.4.2. TU-Member Tier 41
4.5. Performance Comparison and Results 43
4.5.1. Advantages and Performance Comparison for Structure and Design Aspects 43
4.5.2. Advantages of Selected Cryptographic Methods 45
4.6. Conclusions 46
5. THREE-TIER SATELLITE MULTICAST SECURITY MECHANISM BASED ON ECMQV AND IMC METHODS 48
5.1. Introduction to Our Three-Tier Satellite Multicast Security Mechanism 48
5.2. Cryptographic Techniques Used in Our Three-Tier Satellite Security Mechanism 50
5.2.1. The ECMQV Protocol 50
5.2.2. The IMC and IMC Based Methods 51
5.3. Structure and Design Properties of Our Three-Tier Security Mechanism 52
5.3.1. Major Design Properties 52
5.3.2. Structure of Our Security Mechanism 53
5.3.2.1. GEO Satellite Tier 54
5.3.2.2. LEO-MEO Satellite Tier 54
5.3.2.3. TU-Member Tier 55
5.4. Details of Our Three-Tier Security Mechanism 55
5.5. Performance Comparison and Results 60
5.5.1. Performance Comparison of Design Aspects of Our Security Mechanisms 60
5.5.2. Advantages of Cryptographic Methods Used in Our Proposed Security Mechanism 62
5.6. Conclusions 63

6. NAMEPS: N-tier sAtellite Multicast sEcurity Protocol (Mechanism) based on Signcryption schemes 66
6.1. Introduction to NAMEPS 66
6.2. Details of NAMEPS 68
6.2.1. Properties of NAMEPS 68
6.2.2. Detailed Description of NAMEPS 68
6.2.2.1. Initialization, Key Distribution and Data Multicast in GEO Satellite Tier 69
6.2.2.2. Key-Ticket Distribution and Data Multicast in LEO and MEO Satellite Tiers 70
6.2.2.3. Key-Ticket Distribution and Data Multicast in TU Tier 72
6.2.2.4. Data Multicast and Member Management in MMU and Member Tiers 73
6.3. Performance Analysis of NAMEPS 74
6.3.1. Advantages of SCS1M Usage in NAMEPS 74
6.3.2. Advantages of Structural Design and Properties of NAMEPS . . 75
6.3.3. Simulation Results 77
6.4. Conclusions 79
7. HIMUTSIS: Hierarchical MUlti-Tier adaptive ad-hoc network security protocol (mechanism) based on SIgncryption type key exchance Schemes 81
7.1. Introduction to HIMUTSIS 82
7.2. Related Works and Background 84
7.2.1. Cryptographic Techniques Used in Ad-hoc Networks 84
7.3. Structural Design of HIMUTSIS 84
7.4. Cryptographic Techniques and Security Level Structure of HIMUTSIS . 86
7.5. Detailed Description of HIMUTSIS 87
7.5.1. Key Management of HIMUTSIS 87
7.5.2. Detailed Description of HIMUTSIS 88
7.5.2.1. UAV-MBN1 Tier: 89
7.5.2.2. MBN1-MBN2 Tier: 91
7.5.2.3. MBN2-Regular Ground Node Tier: 92
7.6. Performance Analysis of HIMUTSIS 93

7.6.1. Properties of Cryptographic Methods Used in HIMUTSIS: . . . 93
7.6.2. Structural Design and Key Management Properties of HIMUTSIS 94
7.7. Conclusions . . . 96
8. IMC (Improved Merkle Cryptosystem) . . . 98
8.1. Introduction to IMC . . . 98
8.2. MC and VMC . . . 100
8.2.1. Merkle Cryptosystem (MC) . . . 100
8.2.2. Variant of Merkle Cryptosystem (VMC) . . . 102
8.2.3. Advantages of VMC over MC . . . 104
8.2.4. Disadvantages of VMC compared to MC . . . 105
8.3. Improved Merkle Cryptosystem (IMC) . . . 105
8.4. Analysis and Comparison of IMC . . . 107
8.4.1. Security Analysis and Advantages of IMC . . . 108
8.4.2. Storage Analysis and Advantages of IMC . . . 109
8.4.3. Comparison of IMC with MC, VMC and Some Well-known Public Key Cryptosystems . . . 110
8.5. Conclusions and Future Works . . . 112
9. CONCLUSIONS . . . 113
REFERENCES . . . 118

LIST OF FIGURES

Figure 2.1. Structure of the Flat Protocol 11

Figure 2.2. KEKs which are hold by member M4 in a LKH tree 12

Figure 2.3. Initialization phase of join operation for member M3 to LKH tree 12

Figure 2.4. Operations for join event of member M3 to LKH tree 13

Figure 2.5. Operations for leave event of member M3 from LKH tree 13

Figure 2.6. Key requirements of member M4 for to obtained required keys . . 15

Figure 3.1. Comparison of cost of signcryption to cost of sign-then-encrypt . . 24

Figure 4.1. Structure of TTPVSS 47

Figure 5.1. Structure of our satellite security mechanism 64

Figure 6.1. TRBCC comparison of NAMEPS for increasing member size . . . 78

Figure 6.2. TRBCC comparison of NAMEPS for increasing member dynamism 79

LIST OF TABLES

Table 4.1. Workload comparison of TTPVSS to Flat and LKH 47

Table 4.2. Advantages and properties of ECPVSS against its widely used alternatives 47

Table 5.1. The ECMQV Key Establishment Protocol 51

Table 5.2. Performance comparison of our mechanism to Flat, LKH and our previous mechanism (TTPVSS) is given for five major criteria . . . 65

Table 5.3. Comparison of cryptographic protocols with regard to nine essential criteria. RSA-S denotes RSA Signatures and DSA-V denotes DSA Variants 65

Table 7.1. Comparison of HIMUTSIS with some Pure Centralized Key Management Protocol for ORW 96

Table 8.1. Computational and Storage Complexity of MC 102

Table 8.2. Computational and Storage Complexity of VMC 104

Table 8.3. Comparison of IMC to MC and VMC 110

Table 8.4. Comparison of IMC with VMC-MC and some well-known public key cryptosystems for various criteria 111

LIST OF SYMBOLS/ABBREVIATIONS

a	Private key of the signer and is calculated by using I_s and γ
(b, u)	ECPVSS signature pair
(c, r, s)	El-Gamal based Signcryption triplet
(c_i, r_i, s_i) with $term'$	Similar to El-Gamal based Signcryption triplet. In this form, $term'$ with n^{th} degree "$\prime$" denotes modified version of triplets for related tier levels
c_1, c_2, c_3	CWC values for SCS1, TTPVSS and traditional cryptographic approaches respectively in NAMEPS comparison
C	Encrypted part of data in ECPVSS
$CERT_j^{l,i}$	Certificate of the $j'th$ unit in the $l'th$ tier and $i'th$ theater with denoted time intervals for public key $PK_j^{l,j}$
$CRS_{i,j}^{m_1,m_2}$	$(c_{i,j}^{m_1,m_2}, r_{i,j}^{m_1,m_2}, s_{i,j}^{m_1,m_2})$
$CRS_{j,i}^{m_2,m_1}$	$(c_{j,i}^{m_2,m_1}, r_{j,i}^{m_2,m_1}, s_{j,i}^{m_2,m_1})$
d	Hashed ECPVSS value
D_K	Symmetric decryption function where K is the key of decryption function
E	Number of point on Elliptic Curve E
E_K	Symmetric encryption function where K is the key of encryption function
f	Computes functional relationship among node secret
$\mathbb{F}$	Field
g	An integer in [1,...,$p-1$] with order $p-1$ modulo p
G	A public point of order nr in the group of points on elliptic curve $E(\mathbb{F}_q)$ over finite field $\mathbb{F}_q$
h	Secret hashed vector where $h_i \in h$
h'	Encrypted form of h
H	Cryptographic hash function
I_s	Identity of the signer
k	Branching factor of the logical key tree

k_{ind} and k^*_{ind}	When ind is a constant and k_{ind} or k^*_{ind} is individual state then it denotes a KEK in a logical key tree
(k_1, k_2) and (k^*_1, k^*_2)	When $ind = 1, 2$ and k_{ind} or k^*_{ind} is in pair form such as (k_1, k_2) or (k^*_1, k^*_2) then they denotes hashed signcryption key pairs
k_i and k^*_i	When ind in k_{ind} is in variable form then it denotes hashed signcryption value vector in a security mechanism
K	Secret key vector that is used to generate P
$\|K\|$ and $\|K'\|$	Bit length of single key in key management protocols for NAMEPS
KK and KK'	New and previous key of an internal node in ELK tree
$K^{s,d}_{i,j}$	Directed secret key in key exchange procedure. It is transmitted from $i'th$ source s_i to $j'th$ destination d_j
KH_k	Keyed cryptographic hash function
$KT^{\gamma_l}_i$	Intra-theater group communication key generated by theater manager. γ_l represents theater level and index i denotes index of the group manager in level l (l in subscript form)
K_s	Session key
K'_s	Joint session key
l	Number of terrestrial units in satellite multicast system
l' and k'	Hashed signcryption key values
l_t	Number of TU, managed by individual satellite in NAMEPS
m	Number of puzzles in public and private key vectors in MC, VMC and IMC
m'	Base batch keying factor in TTPVSS
m_1, m_2	Batch keying factor in related mechanisms
$m_{i,j}$	Group keys of MMU in NAMEPS
$m_{tiermanager}$	Specific multicast data coming from $tiermanager$ where $tiermanager$ can be $geo, leo - meo$ or TU
mn	Input key bit length of PRF
M	Plaintext data (Bulk multicast data in many case), when M is in form of M' or M'' it denotes encrypted form of M in related tier of networks
M1-constant	M with constant number denotes exemplified member number in logical key trees

n'	Bit length of the secret key for MC
n	Bit length of single puzzle and bit length of the secret key for VMC
n_1, n_2	Key bit length of left and right key part for ELK tree
nk_v	Node key for node nv
nm	Output key bit length of PRF
nv	An interior node in OFT tree
n_l	Average number of members that belongs to a local terrestrial units member group
n_m	Number of member in MMU
n_s	Number of satellites in LEO-MEO satellite tier
n_{tu}	Average number of TUs, which are controlled by a single LEO or MEO satellite in three-tier structure
N	Number of member in secure satellite multicast system
$\overline{N}$	Number of puzzles
p	Large prime number
P	Public key vector (puzzles)
P_i^*	Puzzles generated from y_i
$Pr(Collusion)$	Probability of collusion for random discovery of Bob to find secret key in puzzle P
q	Prime factor of $p-1$
$Q = a \cdot G$	Public key of the signer
r	When r is in individual form, it denotes number of rekeying
rc	Recoverable Commitment Value
rk	Random number used in ECPVSS key generation
(r, s)	El-Gamal based digital signature pair
r_l	Number of rekeying at TU tier in TTPVSS
r_a, r_b	Ephemeral private keys of Alice and Bob in ECMQV
$r_{tierscope}$	Number of rekeying for related tier scope (it may include more than one tier in combined manner)
$\overline{r}_{ind}$	Index number used for key agreement
R_a, R_b	Ephemeral public keys of Alice and Bob in ECMQV
R_i	Receivers in multi-recipient signcryption scheme

R'	Symmetric key derived by using KDF from key R
$s\alpha_i$ and $s\beta_i$	Group key seeds
sm_i	Group key seed for NAMEPS in MMU tier
st_i	Group key seeds in NAMEPS
stc_i	Validation ticket seeds in NAMEPS
$su_{i,j}$	Seed value transmitted from $i'th$ theater manager to $j'th$ node in that theater
sx_i	Seed vectors used in satellite tiers
S	The recognizable redundancy value
t	Bit length of single puzzle
$t_{i,j}$	Group keys generated from st_i seeds
tc_i	Validation tickets generated from stc_i in NAMEPS
$Term_{i,j}^{s,d}$	Similar to $K_{i,j}^{s,d}$. In this notation, superscripts s and d can be u, m_1 or m_2. If u, m_1 and m_2 are in superscript form then they denote UAV, $MBN1$ tier and $MBN2$ tier, respectively
Tr	Bijective transformation
U	Pre-bijective transformation value used as key in Tr
v_i	Random number used in signcryption related steps
V	Plaintext part of data
W_a, w_a	Public and private keys of Alice in ECMQV protocol
W_b, w_b	Public and private keys of Bob in ECMQV protocol
x, x^*	Random number generated by Alice for every El-Gamal signature
x_nv	Node secret for node nv
x_{nl} and x_{nr}	Left and right part key contribution for x_{nv} in OFT
x_a and x_b	Private key of Alice and Bob, respectively
X	The public key value, which used to generate public key vector P
y_a and y_b	Public key of Alice and Bob, respectively
y_i	Auxiliary key
z_i	Group key vector
$z_{i,j}$	Group keys generated from z_i
Z_i	Shared key of Alice and Bob obtained from ECMQV protocol

α_i and β_i	Group key vectors
γ	A point on the curve and is used as implicit certificate
$\|$	Concatenation
$\|var\|$	Bit length of variable var
$\oplus$	XOR
$\vee$	OR
$\wedge$	AND
$O(.)$	Big-Oh Notation for complexity analysis
AES	Advanced Encryption Standard
BBS	Blum-Blum-Shub
BW	Bandwidth
CAS	Classical Asymmetric Cryptography
CBC	Cipher-Block Chaining
CPRNG	Cryptographically secure Pseudo-Random Number Generator
CWC	Cryptographic Workload Coefficient
DES	Data Encryption Standard
DH	Diffie-Hellman Key Exchange Protocol
DKEUN	Direct Key Exchange Using Nonce
DKEUTS	Direct Key Exchange Using Time Stamp
DKEUTSS	DKEUTS parameter transport
DKEUTSV	DKEUTS parameter verification
DKTUN	Direct Key Transport Using Nonce
DKTUTS	Direct Key Transport Using Time Stamp
DLP	Discrete Logarithm Problem
DPM	Digital Post Marking
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
DoS	Denial of Service

ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECMQV	Elliptic Curve Menezes-Qu-Vanstone
ECMQVKG	Static Key Generation for ECMQV Protocol
ECPVSS	Elliptic Curve Pintsov-Vanstone Signature Scheme
EC_Keygen	Elliptic curve key generator that generates and validates ECC parameters
ELK	Efficient Large-Group Key Distribution
FS	Forward Secrecy
GEO	Geostationary Earth Orbit
GK	Group Key
GKMP	Group Key Management Protocol
GSA	Group Security Agent
GSC	Group Security Controller
HAVAL	A one-way Hashing Algorithm with Variable Length of output
HIMUTSIS	HIerarchical MUlti-Tier adaptive ad-hoc network security mechanism based on SIgncryption type key exchance Schemes
HMAC	Keyed Hashing for Message Authentication
HMQV	Hashed Elliptic Curve Menezes-Qu-Vanstone
ICI	Implicit Certificate Information
IMC	Improved Merkle Cryptosystem
KCI-R	Key-Compromise Impersonation Resilience
KDC	Key Distribution Center
KDF	Key Derivation Function
KEK	Key Encryption Key
KKS	Known-Key Security
LB	Lower Bound
LEO	Low Earth Orbit
LKH	Logical Key Hierarchy
LFSR	Linear Feedback Shift Register

MBN	Mobile Backbone Network
MAC	Message Authentication Code
MANET	Mobile Ad-hoc NETwork
MC	Merkle Cryptosystem
MEO	Medium Earth Orbit
MMU	Major Mobile Unit
MR	Message Recovery
MRDS	Message Recovery type Digital Signature
MU	Mobile Unit
NAMEPS	N-tier sAtellite Multicast sEcurity Protocol (Mechanism) based on Signcryption schemes
NTS	Non-Trusted Server
NTRU	Nth degree TRUncated Polynomial Ring
N-Tier	Multi-Tier
OFB	Output Feedback Mode
OFT	One-way Function Tree
ORW	Overall Rekeying Workload
PDA	Pure Decentralized Approach
PKC	Public Key Cryptography
PKCL	Public Key Cryptography bit Length
PRF	Pseudo-Random Function
PRNG	Pseudo-Random Number Generator
RGN	Regular Ground Node
ROM	Random Oracle Model
SCBL	Symmetric Cryptosystem Bit Length
SCS1M	multi-recipient signcryption scheme 1
SDLF	Subgroup Discrete Logarithm Problem Field
SDSS1-2	Shortened Digital Signature Schemes 1-2
SECDSS	Shortened Elliptic Curve Digital Signature Scheme
SGNKG	Signcryption Key Generator
SHA	Secure Hash Algorithm
SKDC	Simple Key Distribution Center

SKG	Symmetric Key Generator
SL	Satellite Tier
$SL\#$	Security Level (SL1-2-3)
SPNG	Strong prime number generator
SPoF	Single Point of Failure
SSMS	Secure Satellite Multicast System
STAKE	Signcryption Type Authentic Key Establishment Protocol
TBOS	Two Birds One Stone
T-Function	Triangular Function
TRBCC	Total Rekeying Bandwidth-Cryptographic Cost
TS	Time Stamp
TTP	Trusted Third Party
TTPVSS	Two-Tier Pintsov-Vanstone Signature Scheme
TU	Terrestrial Unit
UAV	Unmanned Aerial Vehicle
VMC	Variant of Merkle Cryptosystem

1. INTRODUCTION

Wireless communication networks have critical importance in today's communication systems. The rapid growth and development of wireless communication systems into a wide range of networks and for a wide range of applications drives the need of supporting advanced security solutions that meet the requirements of a vast variety of users. On the other hand, wireless networks, having naturally broadcast characteristic, are vulnerable to various passive and active attacks. All the vulnerabilities that exist in conventional wired networks apply to wireless networks. Moreover, broadcast nature of wireless networks makes eavesdropping and active attacks such as message replay, injection and deletion easier. Thus, providing security in wireless networks require usage of advanced cryptographic methods in combined manner. These dense cryptographic processes consume significant amount of computational and storage resources. However, wireless networks are especially resource constraint by means of bandwidth, computational power and energy consumption. Thus, providing high security and performance together is a difficult task in wireless networks [13], [15], [39].

1.1. Secure Group Communication in Wireless Networks

Group communication is one of the essential forms of communication in wireless networks. Thus, multicast applications related to various types of wireless networks are the most common form of group communications. Secure group communication also requires major cryptographic goals such as confidentiality, authentication, integrity, unforgeability and non-repudiation [3]. In secure group communication, in order to secure bulk multicast data, data is encrypted with a group key. All members in the group possess this group key and can decrypt multicast data. Management of group key and other types of keys in multicast system is a very complex task. Details about group key management techniques are given in chapter 2. In this sense, apart from major cryptographic services, secure group communication requires two additional security services: Forward and backward security [5], [70].

Forward and backward security mainly aim to provide the freshness of the cryptographic keys in the group. In order to provide forward and backward security, whenever a member join-leave event occurs, the group key must be updated. In forward security, if a member leaves multicast system, the group key must be changed. Elsewhere the old member still will be able decrypt multicast data even if he is not member of the multicast group. In backward security, if a new member joins to multicast system, the group key must be changed. Elsewhere the new member can store previous data and using newly obtained group key, he can decrypt data even if he was not the member of the multicast group. Notice that, forward and backward security services also need major cryptographic services.

Providing forward-backward security in very large and highly dynamic groups without creating massive workload over all components of the system is a very difficult task without creating massive workload over all components of the system since forward-backward security requires large amount of cryptographic processes and bandwidth consumption together. Notice that, all of these problems can be observed severely in secure satellite multicast systems (SSMS) and Ad-hoc networks. Nowadays, SSMSs gain significant importance especially for pay TV and similar multimedia applications. Satellite systems have the advantage of global coverage and inherent broadcast capability, and offer a solution for providing broadcast access to end users [56], [57]. Also, SSMSs have critical importance for military command control applications. On the other hand, all these satellite networks, having very large number of members and dynamic member behavior characteristic, suffer from cryptographic workload resulting from forward-backward security requirements. Ad-hoc networks have similar problems with SSMS. However, providing security in Ad-hoc networks is a more challenging issue. The reason is that, many SSMSs have semi or full network infrastructure. On the other hand, in Ad-hoc networks, since there is no specific infrastructure, key management is a more difficult task. Moreover, generally, light-weight mobile members are more resource limited that makes providing security more difficult in Ad-hoc networks.

1.2. Content of This Thesis

In this thesis, in order to solve aforementioned problems, we propose various integrated security mechanisms covering large span of cryptographic methods. Our studies focus on three major points in order to solve these problems: Structural design of wireless security systems, hybrid key management techniques and novel cryptographic approaches. Bringing novelties in three major points, we propose generic design structures and principles, which lead the development of seven different studies each of them focusing on various aspects of wireless security mechanisms. Our generic structure offers multi-tiered structures to handle very large groups in hierarchical manner. These structures take into consideration available components of specific wireless network type and needs of these components. Our design principles suggest hybrid key management techniques, which can be used in these multi-tiered network structures. Our hybrid key management approaches integrate logarithmic rekeying cost scaling properties of logical key tree based centralized key management techniques and advantages of decentralized key management protocols such as eliminating single point of failure problems and dividing large group into smaller sub-groups [59]. In this way, our security mechanisms significantly reduce rekeying cost of the forward and backward security requirements. In addition to these, our security mechanisms utilize and adapt various cryptographic methods, which have not been used in SSMS and Ad-hoc networks providing advantages when compared to traditional approaches. Our security mechanisms use Elliptic Curve Cryptography (ECC) based methods [37], message recovery type digital signatures [24], [3], specific authentic key exchange schemes [50] and newly proposed cryptographic methods [23]. Moreover, we especially focus on utilizing signcryption based methods, which have significant advantages when compared to traditional cryptographic approaches.

In chapter 2, we present key management techniques used in our security mechanisms. We give principles and general taxonomy of key management techniques and focus on specific key management techniques that we have utilized in our security mechanisms.

In chapter 3, we present major cryptographic techniques used in our security mechanisms. We briefly mention message recovery type digital signature and authentic key exchange schemes that we have used in our studies. Then, we present shortened digital signatures, which are basis for signcryption schemes. Chapter 3 mainly focuses on signcryption, which is major cryptographic primitive that we have used in our security mechanisms. We give details about basic and multi-recipient signcryption schemes and signcryption based key exchange schemes such as DKEUTS (Direct Key Exchange Using Time Stamp).

In chapter 4, we give details of our TTPVSS (Two-Tier Pintsov-Vanstone Signature Scheme) mechanism. We firstly give description of ECPVSS (Elliptic Curve Pintsov Vanstone Signature Scheme) [47] that is the major cryptographic method of TTPVSS. Using a message recovery type digital signature such as ECPVSS is a novel approach for SSMS which provides at least three times bandwidth gain when compared to nearest traditional cryptographic methods. TTPVSS uses two independent LKH tiers. First tier is satellite-Terrestrial Unit (TU) and second tier is TU-Members tier. Whenever a member join-leave event occurs, only related local TU group is affected from modification. Neither other TUs, nor satellite is affected from modifications. This approach significantly reduces rekeying workload of the satellite. Also, batch keying mechanism is proposed, which reduces rekeying workload of the satellite and provides additional advantages. Performance comparison of TTPVSS to pure implementation of Flat and LKH [7] is given [16].

In chapter 5, we give details of our three-tier satellite multicast security mechanism based on ECMQV (Elliptic Curve Menzes-Qu-Vanstone) [25]and IMC (Improved Merkle Cryptosystem)[23]. We give description of ECMQV and IMC that we used in our security mechanism. Our security mechanism consists of three tiers: GEO satellite tier, MEO-LEO satellite tier and TU-Members tier. GEO satellite tier is mainly responsible for generation of required cryptographic keys and their distribution to whole SSMS. Also, GEO satellite(s) monitors encrypted traffic for security violation. MEO-LEO satellite tier distributes these keys using appropriate algorithms and using satellite internetworking possibilities realizes bulk data multicast to TUs. Each LEO or MEO

satellite manages a TU group and each TU manages a member group. Independency of tiers principle of TTPVSS is also used in these tiers obeying LKH key update rule for member join-leave events. Utilizing advantages of lower delay rate and packet loss of LEO-MEO satellite tier, our three tier mechanism has better batch keying mechanism. In addition to this, NTS (Non-Trusted Servers) are used to handle public keys and certificates for large integrated satellite-terrestrial network. Advantages of ECMQV, ECPVSS and IMC are given together with structural performance gain in performance comparison section [13].

In chapter 6, we give details of our NAMEPS (N-tier sAtellite Multicast sEcurity Protocol (Mechanism) based on Signcryption schemes) mechanism. Details about cryptographic techniques, which are used in NAMEPS, are given in chapter 3. Similar to security mechanisms above, NAMEPS is also designed to handle very large and highly dynamic SSMSs. NAMEPS has four major tier: GEO satellite, LEO-MEO satellite, TU tier and MMU (Mobile Major Unit)-Member tiers. NAMEPS utilizes some principles of TTPVSS and three-tier satellite multicast security mechanism but improves them in many points. Firstly, NAMEPS uses ELK [9] protocol in each of its tier providing member join operation advantages when it is compared to LKH. Also, NAMEPS propose a validation ticket mechanism integrated with batch keying mechanism, which additionally reduces rekeying workload of components of the SSMS. Different from our other SSMS mechanisms, NAMEPS uses multi-recipient signcryption techniques, which provide both computational and bandwidth advantages. Detailed performance comparison and analysis of NAMEPS to pure implementation of some well-known mechanisms are also given [70].

In chapter 7, we give details of our HIMUTSIS (Hierarchical MUlti-Tier adaptive ad-hoc network security protocol based on SIgncryption type key exchange Schemes) mechanism for military MANETs. HIMUTSIS is specifically designed for very large, dynamic and mission critic military Ad-hoc networks. HIMUTSIS also uses independency of tiers principles of our other security mechanisms. However, structural design and cryptographic techniques of HIMUTSIS are completely different from others. As a novelty, in HIMUTSIS, we propose MBN1 (Mobile Backbone Network 1)-MBN2 (Mo-

bile Backbone Network 2) tiers as a new structural approach for MBN tier in military MANETS. This approaches increase networks resistance against single points of failure, reduce delays and most importantly reduce threshold cryptography requirements of the MBN tier. In addition to these, HIMUTSIS uses multi-level security system having different key bit length and cryptographic algorithm for different tiers in order to optimize security performance tradeoff in real-time digital battle field environments. In HIMUTSIS, we adapt SDSS-1 (Shorted Digital Signature 1) signcryption based DKEUTS protocol which provides high security for both passive and active attacks and give advantages for computational and bandwidth consumption aspects [59].

In chapter 8, we give details of our IMC (Improved Merkle Cryptosystem) algorithm. In this chapter, different from other studies, we focus on improving an existing cryptography algorithms, MC (Merkle Cryptosystem), and improve MC and VMC (Variant of Merkle Cryptosystem) for both security and performance aspect. IMC uses different puzzle structure from MC and uses cryptographic hash functions in order to improve VMC. In this way, IMC can provide a security level, which is competitive for today's many modern public key cryptosystems. Apart from IMC, we mention STAKE (Signcryption Type Authentic Key Establishment Scheme), which is under preparation. STAKE utilities principles of signcryption based key establishment schemes and IMC to create an authentic key exchange protocol [23].

Notice that, we give details of our five major studies in this thesis. We will not give details of [1] and [78] but present their properties and functions for our major studies.

2. KEY MANAGEMENT TECHNIQUES USED IN OUR MECHANISMS

Key management is one of the most important issues in security mechanism design. In our security mechanisms, we especially focus on group key management protocols and propose new hybrid key management methods for various wireless network systems such as satellite networks and military mobile Ad-hoc networks. Notice that, if an appropriate key management technique is not used then neither advanced cryptographic techniques nor structural design of the network system can be sufficient to provide security.

In this chapter, we give major cryptographic techniques which are specifically used in our security mechanisms. Firstly, we mention general properties and principles of group key management protocols. Then, we give general taxonomy of key management protocols and present details of some centralized and decentralized key management protocols that we have utilized in our security mechanisms. Notice that, we have combined these techniques in hybrid key management techniques so that they can show high performance for many metrics when compared to traditional approaches.

2.1. Principles of Key Management Protocols for Secure Group Communication

In group communication, in order to provide appropriate distribution and maintenance of cryptographic keys, key management techniques are used as an essential method. Multicast applications are the most common form of the group communication. In multicast communication, a central entity transmits same messages to a group of member. Bulk data multicast applications are the most common form of the multicast. Thus, symmetric cryptography based approaches are generally preferred in order to encrypt bulk multicast data. However, in order to do this, symmetric keys must be distributed among members appropriately. Hybrid cryptograph techniques are used together with key management protocols to achieve desired cryptographic goals.

Group Key (GK) is used to encrypt bulk multicast data. Every member in group knows GK and can decrypt multicast data using GK. However, GK must be transmitted to members securely. Generally, a key exchange scheme or other public key cryptography based methods are used for this purpose. The cryptographic keys, which are used to encrypt GK, are called as Key Encryption Key (KEK). Using appropriate cryptographic method, each member obtains KEK and can decrypt GK. Thus, key management problem can be considered as secure and efficient distribution of KEKs and GK to only valid members in the group. However, generally multicast systems have large and dynamic groups in which frequent member join-leave events occurs. Consequently, a key management protocol must be able to handle cryptographic workload resulting from very large and dynamic structure of the group and provide freshness of the cryptographic key in the network [1].

In large multicast systems, most costly operation is the rekeying operation. Rekeying is done in order to provide freshness of the cryptographic keys in the group. Main purpose of the rekeying operation in group communication is providing forward and backward security [2]. Notice that, forward-backward security requirements are different from classical cryptographic goals such as confidentiality, authentication, unforgeability and non-repudiation [3]. In order to provide forward and backward security, the GK must be changed for each membership change. In backward security, whenever a member join occurs to group, the GK must be updated. If group key is not updated, then new member can decode previous messages before it joins the group. To do this, candidate member consciously records previous encrypted messages. However, if the GK is updated then member can not decrypt previous messages using new GK. In forward security, whenever a member leaves from the group, the GK must be updated. If the GK is not updated then old member can still decrypt the current messages of the group since it knows group key. Changing the GK for member leave operation, old member can not monitor messages of the group [4].

Providing forward and backward security in a large and dynamic group causes massive workload over system. Consider a member group having thousands of members many of them are mobile members. This group tends to be highly dynamic and member

join-leave events frequently occur. Under this condition, sending a new key to each member one-by-one is not feasible for each member join-leave event [5].

2.2. Analysis of Key Management Techniques Used in Our Mechanisms

In our security mechanisms, we have utilized many different key management protocols in order to create efficient and network structure-suit key management approaches. For this reason, in this section, we firstly give general taxonomy of the key management protocols.

Various key management protocols have been proposed having different properties. Mainly, key management protocols can be divided into three major categorizes [6]: Centralized key management protocols, decentralized key management protocols and hybrid key management protocols. Centralized key management protocols generally use a logical key tree structure for cryptographic keys such as LKH (Logical Key Hierarchy) [7], OFT (One-way FunctionTree) [8], and Efficient Large-Group Key distribution (ELK) [9]. Decentralized key management protocols divide large group into subgroups and manage each subgroup individually such as Iolus [10] and Kronos [11]. Also, hybrid approaches exist such as Mykil [12]. Notice that, our mechanisms TTPVSS, three-tier satellite security mechanism, NAMEPS and HIMUTSIS also use hybrid key management approaches. Details of the key management approach of each mechanism are given in their own sections.

In this section, we firstly give details of centralized key management protocols that we have used in our security mechanisms. Then we describe decentralized key management protocols that we have combined with centralized key management protocols in our security mechanisms.

2.2.1. Centralized Key Management Protocols

In centralized key management protocols, only single central entity controls the whole group. Centralized key management techniques have both advantages and disad-

vantages for various factors. Major problem in centralized key management protocols is Single Point of Failure (SPoF) problem. If central entity becomes unavailable then the complete key management service will not function. When central entity is not active, members of the central entity are under security threats. Also, if central entity is compromised then all cryptographic keys, certificates etc. about the whole group are also compromised.

However, centralized key management protocols are well-suited for many real-life applications. Especially military applications and many of the civilian applications have naturally hierarchical and central entity based structure. Also, there are many efficient centralized key management protocols, which can handle very large and dynamic groups in logarithmical key management cost. This low-cost property makes central key management protocols an attractive security approach. An efficient key management protocol aims to minimize storage requirements of both central entity and members. Other critical goals are minimizing rekeying cost and size of the messages transmitted over network.

In next subsections, we describe Simple Key Management Protocol (Flat), LKH, OFT and ELK protocols, which we have utilized in our security mechanisms.

2.2.1.1. Simple Key Management Protocol. The most straightforward centralized key management technique is Simple Key Management Protocol or Simple Key Distribution Center (SKDC). SKDC is also called as Flat protocol. In Flat system, each member directly is connected to the group manager and a unique KEK is assigned to a member. Whenever a key update occurs, the GK is sent to each member one by one encrypting it by unique keys of each member. Thus, key update cost of the Flat system is N where N is the number of members in the group [13]. Notice that, Flat system can not handle large and dynamic groups having more than a few thousand members. However, using Flat system for some specific task is possible. For instance, in [1], we have used Flat protocol in satellite-TU tier. Since number of TUs are limited, application of Flat protocol is feasible. Also, some principles of Flat protocol is used in Group Key

Management Protocol (GKMP) [14]. Figure 2.1 shows structure of the Flat protocol.

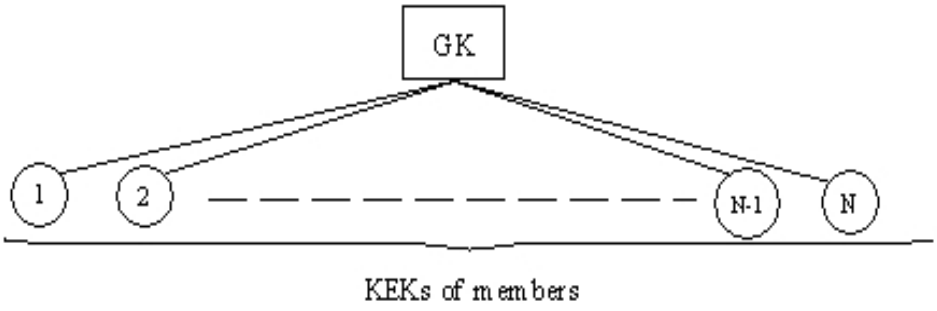


Figure 2.1. Structure of the Flat Protocol

2.2.1.2. Logical Key Hierarchy (LKH). Logical Key Hierarchy (LKH) is one of the most fundamental central key management protocols that leads the invention of many other logical key tree based methods. We use LKH protocol in our [1], TTPVSS [16] and Three-tier satellite multicast security mechanism [13]. LKH was proposed by Wallner at al. [7] in 1999.

In LKH, central manager holds a logical key tree, each of its nodes are KEKs. The leaves of the logical key tree corresponds to group members. Each member stores a key vector to reach GK. This key vector contains KEKs which take place on the path of the member to reach the root of the logical key tree. The key at the root is GK. Figure 2.2 shows a simple logical key tree which is used for a group having eight members. Member four (M4) holds the following key vector $\{k_{11}, k_5, k_2, k_1\}$. In this way, for each member join leave event, only keys that are on the affected paths are updated. This approach reduces rekeying cost of the LKH N to $k{\cdot}log_k N$ where k is the branching factor of the logical key tree (when k is used in centralized key management rekeying cost, it represents branching factor). Height of the logical key tree is $log_2 N$.

In member join events, all KEKs in the nodes for the new leaf's parent in the path to the root are compromised and should be changed. Thus, rekeying cost of the LKH for $k = 2$ is $2log_2 N$ keys long. Figure 2.3 and Figure 2.4 shows an example of member join operation. Suppose that a new member, M3 wants to join to the group. Firstly, group manager sends k_{10} to the M3 and M3 is inserted to the node k_5. Now, from

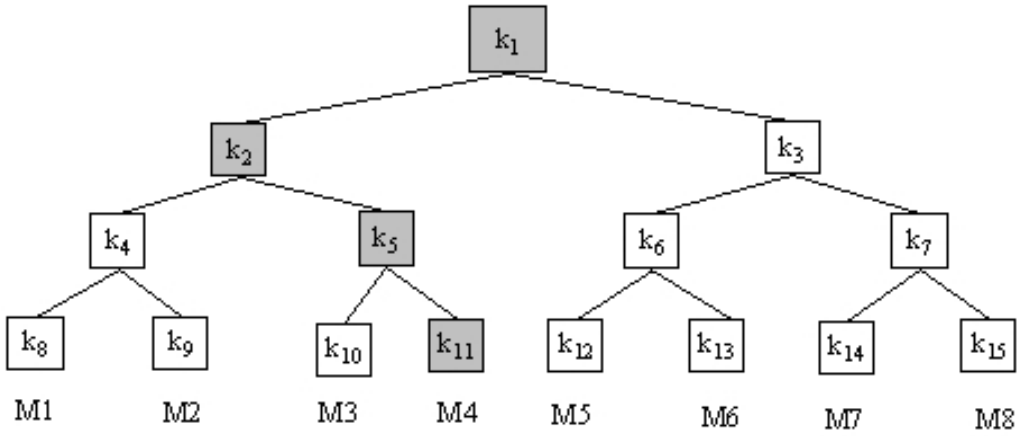


Figure 2.2. KEKs which are hold by member M4 in a LKH tree

M3 to the root, KEK path k_5, k_2 and k_1 must be updated. Group manager generates k_5^*, k_2^*, and new GK k_1^*. Then, following KEK encryption and transmission sequences are performed: k_1^* is encrypted with k_3 and is sent to the right part of the tree. k_1^* is encrypted with k_2^* and is sent to the left part of the tree. k_2^* is encrypted with k_4 and is sent to sub-left part of the tree. Using k_4, members can decrypt k_2^*. Also, new KEK k_2^* is encrypted with k_5^* and is sent to sub-right part of the tree. k_5^* is encrypted with k_{10} and k_{11} and is sent to M3 and its siblings. The size of the rekeying message of a balanced tree has at most $2log_2N$ keys.

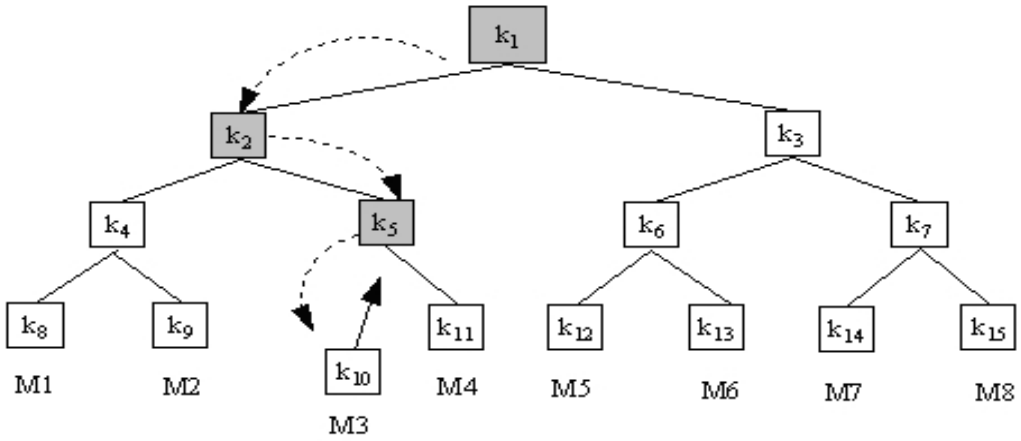


Figure 2.3. Initialization phase of join operation for member M3 to LKH tree

Member leave operation is similar to member join operation. Figure 2.5 shows an example of member leave operation. Suppose that, member four (M4) wants to leave from the group. M4 knows k_5, k_2, and k_1 and these keys must be updated. Group manager generates new k_5^*, k_2^* and group key k_1^*. Following operations are performed:

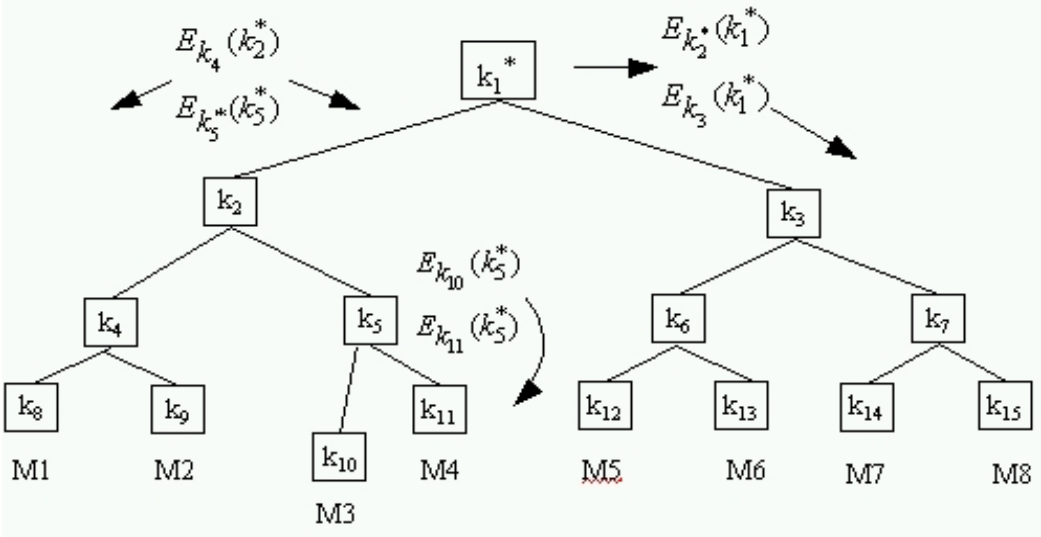


Figure 2.4. Operations for join event of member M3 to LKH tree

At the root of the tree, $E_{k_2^*}(k_1^*)$, $E_{k_3}(k_1^*)$, second level of the tree, $E_{k_4}(k_2^*)$, $E_{k_5^*}(k_2^*)$, third level of the tree, $E_{k_{10}}(k_5^*)$ [5].

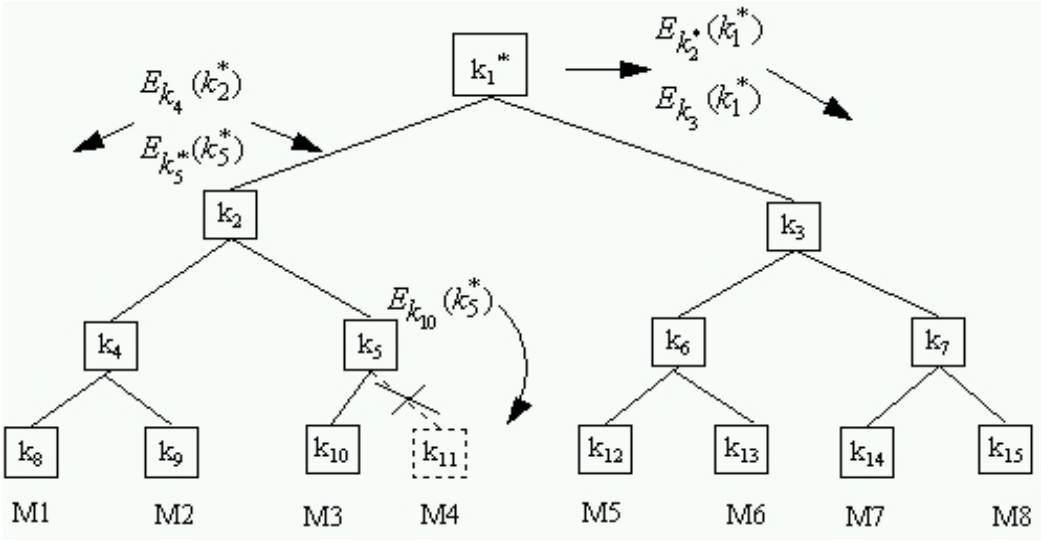


Figure 2.5. Operations for leave event of member M3 from LKH tree

Apart from LKH, a slightly improved version of LKH, LKH+ is proposed in [17].

2.2.1.3. One-Way Function Tree (OFT). In HIMUTSIS, we utilize OFT [8] as a major key management protocol. This group key management protocol uses a special one-way function to compute a tree of keys, which is called One-way Function Tree algorithm. The keys are computed bottom-up fashion, from the leaves to the root. This approach reduces rekeying broadcast to only about $logN$ keys. For instance, OFT can handle a

group with 10 million members with 23 message units for a member leaving operation. When compared to LKH, which uses top-down logical key hierarchy approach, bottom-up approach of OFT halves the required broadcast size. Notice that, unlike LKH, group members contribute the entropy of the GK in OFT. OFT can provide forward and backward security and can be used for both unicast and multicast purposes.

In OFT, group manager holds a binary tree, each node has two types of cryptographic keys: Node secret x_{nv} and a node key nk_v. The node secrets are functionally related by means of a special one-way function. Before giving more details, we give function types which are used in OFT.

Three types of cryptographic functions are mainly used in OFT. In order to provide confidentiality of messages, a symmetric encryption function $E_{k_{ind}}(M)$ is used. In OFT, two special one-way functions are used. The function f is used to compute functional relationship among node secrets. The function g is used to compute each node key from its corresponding node secret. Both f and g compute "blinded" values from the node secrets in such a way that protects the confidentiality of node secrets, for any node secret.

In OFT, the secrets are blinded in the sense that a computationally limited adversary known $f(x_{nv})$ and yet can not find x_{nv}. Similarly, for each node, the node key is computed from the node secret using g, thus, $nk_v = g(x_{nv})$. Now, let nv be any interior node in the key tree, and L and R is the left and right child of nv. In order to determine group keys, OFT uses all of the leaf secrets in the key tree. The group key is computed as a tree of function computations going from the leaf nodes to the root. Specifically, the node secret nv is computed by $x_{nv} = f(x_{nl}) \oplus f(x_{nr})$.

Main security properties of the OFT is following: Each member knows the node secret on the path from its node to the root (the node keys along this path), and blinded node secrets that are siblings to this path, and no other secrets nor node keys [18].

In figure 2.6, we represent an example about key possession of a member in a simple OFT system. Member four (M4) knows k_{11}, blinded value of k_{10}, k_{11} and k_3. Using these keys in $x_{nv} = f(x_{nl}) \oplus f(x_{nr})$ formula, M4 can generate all keys in its ancestor path $\{k_5, k_2$ and $k_1\}$.

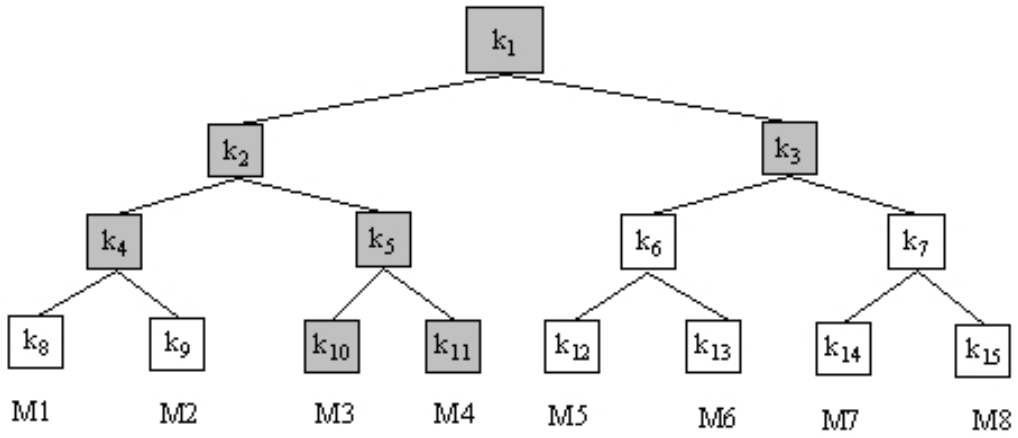


Figure 2.6. Key requirements of member M4 for to obtained required keys

2.2.1.4. Efficient Large-Group Key distribution (ELK). Efficient Large-Group Key distribution (ELK) is proposed by Perrig et al. [9]. ELK has similar properties to OFT in the sense that apparent node key is generated from its children. However, ELK uses PRF (Pseudo-Random Functions) [19]. These PRFs use a key, takes mn bit input and gives nm bit output. Mainly, there are four PRFs, which are used to derive different cryptographic keys in ELK. A new key KK' of an internal node is generated from its previous key KK, n_1 bits contributes from its left child and n_2 bits contributes from its right child where $n_1 + n_2 \leq n$. Notice that, ELK requires higher computational cost at group members to reduce the communication cost [6]. However, ELK has many advantages when compared to some other centralized key management protocols.

Firstly, ELK provides super-efficient member join when compared to many other protocols. For single member join, ELK does not cause rekeying workload. In addition to this, ELK uses smaller key update messages than previous protocols. ELK allows short hint messages to be used for key recovery allowing a tradeoff of communication overhead with member computation [9].

2.2.2. Decentralized Key Management Protocols

In decentralized key management protocols, the large group is divided into smaller subgroups. Each subgroup is managed by a different central entity. This approach aims to eliminate single point of failure problem in centralized key management protocols. A decentralized key management protocol focuses on following criteria:

If past keys are compromised, current keys should not be affected from this problem. One of the most important advantages of decentralized structures is local rekeying possibility. In local rekeying, whenever a rekeying occurs, effects of the rekeying operation are restricted in its local group. Other subgroups are not affected from modification. In addition to these, data path should be independent from the key management path. The reason is that, rekeying operation should not cause any delay over data communication.

In the following section, we give brief description of Iolus [10]. Iolus is the main dynamic decentralized protocol which we utilize its local rekeying properties in our security mechanisms.

2.2.2.1. Iolus. Iolus [10] utilizes decentralized key management approach and especially provides local rekeying properties. In Iolus, large group is divided into subgroups and each subgroup is managed by a Group Security Agent (GSA). Also, GSAs compose another level of group which is managed by a Group Security Controller (GSC).

Each GSA uses different group key for key management in their local group. Thus, if a member join-leave event occurs in a subgroup, rekeying is done only its subgroup and other subgroups are not affected from this modification. This approach provides scalability especially for managing very large and dynamic groups. In addition to these, single point of failure problem does not arise in this structure. Main problem of Iolus is that translation of messages from one group to another group causes delay problems.

3. CRYPTOGRAPHIC TECHNIQUES USED IN OUR SECURITY MECHANISMS

In our security mechanisms, we utilize and adapt many different cryptographic techniques which provide significant security and performance advantages when compared to traditional cryptographic approaches. We utilize large span of public key and hybrid cryptography techniques in order to achieve major cryptographic goals and provide additional security properties with high performance in wireless networks. We mainly use different types of key transport and key exchange protocols to securely transmit KEKs and GK to participants of the communications in group key management. Using these keys, bulk data multicast and other dense communication can be done using symmetric cryptography. This hybrid cryptography approach is an essential technique, which is used together with key management protocols that we have mentioned in the previous chapter.

The major cryptographic technique that we use in our security mechanisms is signcryption. Signcryption is a relatively new concept in cryptography, which was proposed by Y. Zheng [20]. Signcryption combines public key cryptography and symmetric key cryptography in an efficient manner so that it provides many advantages when compared to classical sign-then-encrypt approach. We give details of signcryption and signcryption based methods, which we have used in our security mechanisms, in the following sections.

We have used multi-recipient signcryption [21] in NAMEPS as a major cryptographic technique. Details are given in chapter 6. Moreover, we have used SDSS1-2 (Shortened Digital Signature Schemes 1-2) based DKEUTS-DKEUN (Direct Key Exchange Using Timestamps - Direct Key Exchange Using Nonce) [22] in HIMUTSIS as a major cryptographic technique. Details are given in chapter 7. As a novelty, in NAMEPS, we have adapted multi-recipient signcryption scheme to our N-tier SSMS (Secure Satellite Multicast System). In HIMUTSIS, as a novelty, we have adapted one-

to-one SDSS1-2 based DKEUTS-DKEUN scheme to our multi-tier military MANET and hybrid key management method modifying DKEUTS to m-to-n key exchange protocols.

In addition to these, as a future work, we consider utilizing principles behind the signcryption schemes and integrating these principles with our IMC [23] algorithm. We call this approach as the Signcryption Type Authentic Key Establishment protocol (STAKE). In STAKE, as a novel approach, we will try to combined SDSS1-2 based DKEUTS-DKEUN and our IMC [23] method.

Apart from signcryption based techniques, we utilized and adapted many different and efficient key transport and exchange schemes based on ECC signature schemes. Firstly, as a novel approach, we have used Elliptic Curve Pintsov-Vanstone Signature Scheme (ECPVSS) [24] as a key transport method in SSMS. ECPVSS has been used for Digital Post Marking (DPM) applications for bandwidth constraint environments. However, as far as our concern, ECPVSS has not been used to securely transmit KEKs in secure group communication. ECPVSS, as a MRDS (Message Recovery Type Digital Signature) [3], provides many advantages when compared to traditional approaches in SSMS. We give details of usage of the ECPVSS in the TTPVSS in chapter 4.

In our Three-tier Satellite Multicast Security Mechanism [13], as a novel approach, we have used Elliptic Curve Menezes-Qu-Vanstone (ECMQV) [25] as the major cryptographic technique. Additionally, we suggest using IMC and IMC based techniques together with ECMQV in SSMS. ECMQV is one of the most efficient authentic key exchange protocol which uses principles of ECC and DH in integrated manner. We give details of usage of the ECMQV in our Three-tier Satellite Multicast Security Mechanism in chapter 5.

In next section, we firstly describe SDSS1 and SDSS2 which are frequently used in signcryption schemes. Then, we give description of signcryption schemes in detail. Following two sections, we present some signcryption based key establishment protocols which are used in our security mechanisms such as multi-recipient signcryption and

DKEUTS protocol.

3.1. Shortened Digital Signature Schemes (SDSS)

One of the most important digital signature schemes is ElGamal digital signature scheme [27]. Similar to Digital Signature Algorithm (DSA) [2], ElGamal signature scheme is also based on Discrete Logarithm Problem (DLP). Thus, security of the ElGamal signature scheme is based on the intractability of the DLP. Notations and brief description of ElGamal signature is given below:

p : Large prime number,

g : An integer in [1,...,$p-1$] with order $p-1$ modulo p.

x_a : Private key of Alice. It is randomly chosen from [1,...,$p-1$] with $x_a \nmid (p-1)$.

y_a : Public key of Alice. $y_a = g^{x_a} \bmod p$.

Signature of the message of Alice M is the signature pair (r, s) :

1) $r = g^x \bmod p$.

2) $s = (H(M) - x_a \cdot r)/x \bmod p$.

In this description, H is a one-way cryptographic hash function such as a member of SHA family [28] or HAVAL [29]. x is chosen independently at random from [1,...,$p-1$] with $x \nmid (p-1)$ every time a message is to be signed by Alice. Given (M, r, s), one can verify signature by checking $g^{H(M)} == y_a^r \cdot \mathrm{r}^s \bmod p$ is satisfied.

Shortened Digital Signature Schemes (SDSS) are improved version of ElGamal signature schemes. Details of the required modifications for ElGamal signature to create SDSS can be found in [30]. Brief descriptions of SDSS1 and SDSS2 are given

below:

A. SDSS1:

Signature Steps:

A.1) $r = H(g^x \bmod p, M)$,

A.2) $s = x/(r + x_a) \bmod q$.

Verification Steps: $k' = (y_a \cdot g^r)^s \bmod p$ check whether $H(k', M) == r$

B. SDSS2:

Signature Steps:

B.1) $r = H(g^x \bmod p, M)$,

B.2) $s = x/(1 + x_a \cdot r) \bmod q$.

Verification Steps: $k' = (g \cdot y_a^r)^s \bmod p$ check whether $H(k', M) == r$

Notice that, SDSS1 is slightly more efficient than SDSS2 in signature generation, as the latter involves an extra modular multiplication. Length of the signature for both SDSS1 and SDSS2 is $(|H(\cdot)| + |q|)$. Both SDSS1 and SDSS2 provide advantages when compared to traditional DSS. Their signature sizes are shorter: $2\,|q|$ bits for DSS, while $(|hash(\cdot)| + |q|)$ bits for SDSS1 and SDSS2. Modular inversion or division is not required in signature verification. They both admit provable security, albeit in the random oracle model (ROM). As security properties, SDSS1-2 are unforgeable by adaptive attacker, under the assumptions that DLP is hard (with respect to g chosen uniformly at random and that the one-way hash function behaves like a random function).

3.2. Signcryption Schemes

Signcryption is a relatively new concept in cryptography. Signcryption scheme is a cryptographic method that fulfills both the functions of secure encryption and digital signature, but with a cost smaller than that required by signature-then-encryption [31]. It consists of a pair of (polynomial time) algorithms one of them is signcryption algorithm, generally it is probabilistic, and other is unsigncryption algorithm generally it is deterministic. These algorithms should satisfy following conditions: Unique unsigncryptability, security and efficiency. Many efficient signcryption schemes have been proposed. Shortened El-Gamal signature based signcryption is one of the most characteristic signcryption scheme [20]. It uses El-Gamal variants [32] and shorted signatures such as SDSS1-2 to create digital signcryption schemes. Many additional properties such as name binding, past recovery, forward secrecy and repudiation settlements can be included in signcryption schemes. Apart from computational advantages, signcryption schemes provide communication overhead advantages [21]. Other important extensions of signcryption scheme is based on SECDSS1-2 (Shorted Elliptic Curve Digital Signature Standard 1-2). Notice that, DLP based signcryption schemes can be easily extended to the elliptic curve cryptography domain. In [33], an efficient signcryption scheme, that is based on SECDSS, is given. SECDSS provides approximately 58% computational cost saving and 40% communication overhead saving in average compared to the classical digital signature approaches. Another well-known signcryption scheme is TBOS (Two Birds One Stone) [34] that is based on RSA. Formal proofs for the security of signcryption schemes are given in [35].

In our security mechanisms, we use El-Gamal based signcryption schemes. SECDSS based signcryption scheme has nearly same steps. Details of signcryption scheme can be found in [20]. We give only the description of El-Gamal based signcryption scheme that is the basis for DKEUTS-DKEUN and multi-recipient signcryption schemes which we have used in our security mechanisms.

Public Parameters for all:

p : Large prime number,

q : A large prime factor of $p-1$.

g : An integer in [1,...,$p-1$] with order $p-1$ modulo p.

Parameter of Alice:

x_a : Private key of Alice. It is randomly chosen from [1,...,$p-1$] with $x_a \nmid (p-1)$.

y_a : Public key of Alice. $y_a = g^{x_a} \bmod p$.

Parameter of Bob:

x_b : Private key of Bob. It is randomly chosen from [1,...,$p-1$] with $x_b \nmid (p-1)$.

y_b : Public key of Bob. $y_b = g^{x_b} \bmod p$.

A. Signcryption of Message M by Alice (the sender):

1. Alice select rc at random from [1,...,$q-1$] and computes $l' = H(y_b^{rc} \bmod p)$. Split l' into l'_1 and l'_2 approptiate length.

2. $r = H(M, bind_info, l'_2)$, bind_info contains data that identify the sender such as a public key certificate. l'_2 is the key of keyed cryptographic hash function.

3. $s = rc/(r + x_a) \bmod q$ if SDSS1 is used and $s = rc/(1 + x_a \cdot r) \bmod q$ if SDSS2 is used.

4. $c = E_{l'_1}(M)$.

5.Alice sends the signcrypted text as (c, r, s) triplet to Bob.

B. Signcryption of (c, r, s) by Bob (the recipient):

1. Bob recovers l' from s, r, g, p, y_a, x_b :

$l' = H((y_a \cdot \mathrm{g}^r)^{s \cdot x_b} \bmod p)$ where $s = rc/(r + x_a) \bmod q$ if SDSS1 is used and

$l' = H((y_a^r \cdot \mathrm{g})^{s \cdot x_b} \bmod p)$ where $s = rc/(1 + x_a \cdot r) \bmod q$ if SDSS2 is used.

2. Split l' into l'_1 and l'_2 in appropriate length.

3. $M = D_{l'_1}(c)$.

4. $r' = H(M, bind_info, l'_2)$.

5. if $(r == r')$ then M is a valid message originated from Alice else M is not a valid message.

The main principle behind the El-Gamal based signcryption schemes is to use recoverable commitment value. If a sender computes the commitment value in a special way so that it is only recoverable by an intended receiver, then commitment value can be used as a symmetric key shared between the sender and the receiver and so symmetric encryption can be applied for providing message confidentiality. In the signcryption scheme above, r and s values are used to recover l that contains symmetric encryption key and the key that is used for keyed hash function [29]. Notice that, as a future work, we consider to use this principle to design a new key establishment protocol, STAKE, which will be based on signcryption type approach and IMC [23].

Comparison of cost of signcryption to cost of signature-then-encryption is given in figure below [30].

Various schemes	Computational cost	Communication overhead (in bits)
signature-then-encryption based on RSA	EXP=2, HASH=1, ENC=1 (EXP=2, HASH=1, DEC=1)	$\|n_a\| + \|n_b\|$
signature-then-encryption based on "DSS + ElGamal encryption"	EXP=3, MUL=1, DIV=1 ADD=1, HASH=1, ENC=1 (EXP=2.17, MUL=1, DIV=2 ADD=0, HASH=1, DEC=1)	$2\|q\| + \|p\|$
signature-then-encryption based on "Schnorr signature + ElGamal encryption"	EXP=3, MUL=1, DIV=0 ADD=1, HASH=1, ENC=1 (EXP=2.17, MUL=1, DIV=0 ADD=0, HASH=1, DEC=1)	$\|hash(\cdot)\| + \|q\| + \|p\|$
signcryption SCS1	EXP=1, MUL=0, DIV=1 ADD=1, HASH=2, ENC=1 (EXP=1.17, MUL=2, DIV=0 ADD=0, HASH=2, DEC=1)	$\|KH.(\cdot)\| + \|q\|$
signcryption SCS2	EXP=1, MUL=1, DIV=1 ADD=1, HASH=2, ENC=1 (EXP=1.17, MUL=2, DIV=0 ADD=0, HASH=2, DEC=1)	$\|KH.(\cdot)\| + \|q\|$

where
EXP = the number of modular exponentiations (a fractional number indicates an average cost),
MUL = the number of modular multiplications,
DIV = the number of modular division (inversion),
ADD = the number of modular addition or subtraction,
HASH = the number of one-way or keyed hash operations,
ENC = the number of encryptions using a private key cipher,
DEC = the number of decryptions using a private key cipher,
Parameters in the brackets indicate the number of operations involved in "decryption-then-verification" or "unsigncryption".

Figure 3.1. Comparison of cost of signcryption to cost of sign-then-encrypt

3.3. Achieving Major Cryptographic Goals in Key Establishment Protocols

Key establishment is one of the most challenging issues in cryptography. It is possible to classified key establishment protocols such as key exchange protocols and key transport protocols. In key exchange protocols, session key is obtained by joint keys from both participants. Most well known key exchange protocol is Diffie-Hellmann (DH) [36] key exchange protocol. DH uses DLP problem to provide commutative behavior of exponentiation functions and realize key exchange for parties of communication. In key transport protocols, session keys are generated from one party and these keys are securely transmitted to the other parties of communication [22]. For some applications, key exchange protocols can be preferred due to both parties of communications involve to determine session key. Note that, key exchange protocol and key agreement protocol terms can be used interchangeably. A key establishment protocol must provide security and authentication:

Security: Session keys should be known by only to the participants of communication and a TTP (Trusted Third Party) like a KDC (Key Distribution Center) if it is necessary. Possible attacks for key establishment protocols are inferring session key with passive attacks, replay attacks, interleaving attacks and deducing session key with a known past session key.

Authentication: In authentication, a participant of communication is convinced of the identity of another participant. Authentication can be unilateral or mutual. A protocol that offers both key establishment and authentication is called authenticated key establishment. Symmetric cryptosystem based protocols provide authentication using challenge-response mechanism. Public cryptosystem based protocols provide authentication using digital signatures. Digital signature is a cryptographic primitive which is fundamental in authentication, integrity, unforgeability and non-repudiation [3]. A typical digital signature uses public key cryptography to transmit signatures of messages over insecure channel and uses cryptographic hash functions to obtain signatures. DSA based on El-Gamal cryptosystem, ECDSA (Elliptic Curve Digital

Signature) [37] are well known examples of digital signatures.

Participants of communication may have a shared secret key, which is previously known by participants. In this case, using symmetric key cryptography is more efficient than public key based methods. If participants of communication do not have a previously known secret key, then DH like approaches can be used. However, in order to achieve major cryptographic goals together, hybrid cryptography approaches, which integrate advantages of symmetric cryptosystems and public key cryptosystems, are preferred [38]. As we have described above, signcryption schemes are one of the most efficient hybrid cryptography approaches having advantages over classical hybrid cryptography methods. In our studies, we use advantages of signcryption based key establishment protocols in our wireless security mechanisms. Next section gives descriptions of these schemes.

3.4. Signcryption Based Key Establishment Protocols Used in Our Mechanisms

In this section, we give description of SDSS1-2 based DKETUN, DKTUTS, DKEUTS cryptographic protocols and multi-recipient signcryption scheme that we have used in HIMUTSIS and NAMEPS, respectively.

In order to provide freshness in key transport and exchange protocols, time-stamps or nonce can be used. Also, it is possible to transmit session key directly or indirectly. In direct signcryption based key transport protocols, a session key is encrypted as a message and transmitted in encrypted form with parameter c. In indirect key transport protocol, one of the parts of the hashed key value is used as a session key. In these protocols, etc contains some recognizable value such as identity of the participants or a public key certificate. KH_k denotes a keyed cryptographic hash function and TS denotes time-stamp. Description of DKTUN, DKTUTS and DKEUTS are below [30].

A. Direct Key Transport Using a Nonce (DKTUN):

1. *Bob:* Bob generates and sends a nonce to Alice. $NC_b \in_R \{0,1\}^{l_n}$.

2. *Alice:* Alice generates a session key and sends it to Bob by signcrypting it.

2.1. $key \in_R \{0,1\}^{l_n}$,

2.2. $x \in_R [1, ..., q-1]$,

2.3. $(k_1, k_2) = H(y_b^x \bmod p)$,

2.4. $c = E_{k_1}(key)$,

2.5. $r = KH_{k_2}(key, NC_b, etc)$,

2.6. $s = x/(r + x_a) \bmod q$,

2.7. Alice sends (c, r, s) to Bob.

3. *Bob:* Bob verifies (c, r, s) with unsigncryption. Then optionally, Bob sends a tag to Alice for secondary verification.

3.1. $(k_1, k_2) = H((y_a \cdot \mathrm{g}^r)^{s \cdot x_b} \bmod p)$,

3.2. $key = D_{k_1}(c)$,

3.3. If $KH_{k_2}(key, NC_b, etc) == r$ then message is valid else reject.

3.4. $tag = MAC_{key}(NC_b)$.

4. *Alice:* Alice verifies tag.

B. Direct Key Transport Using a Time-Stamp (DKTUTS):

1. *Alice:* Alice generates a TS and signcrypt TS together with session key.

1.1. key$\in_R \{0,1\}^{l_k}$,

1.2. x$\in_R [1, ..., q-1]$,

1.3. $(k_1, k_2) = H(y_b^x \bmod p)$,

1.4. Get current time-stamp TS,

1.5. $c = E_{k_1}(key, TS)$,

1.6. $r = KH_{k_2}(key, TS, etc)$,

1.7. $s = x/(r + x_a) \bmod q$,

1.8. Alice sends (c, r, s) to Bob.

2. *Bob:* Bob verifies (c, r, s) with unsigncryption. Then optionally, Bob sends a tag to Alice for secondary verification.

2.1. $(\mathrm{k}_1, k_2) = H((y_a \cdot \mathrm{g}^r)^{s \cdot x_b} \bmod p)$,

2.2. $(key, TS) = D_{k_1}(c)$,

2.3. $\mathrm{KH}_{k_2}(key, TS) == r$ then message is valid else reject.

2.4. $tag = MAC_{key}(TS)$.

3. *Alice:* Alice verifies tag.

C. Direct Key Exchange Using Time-Stamp(DKEUTS):

1. *Alice:* Alice generates a TS and signcrypt TS together with session key.

   1.1. $key \in_R \{0,1\}^{l_k}$,

   1.2. $x \in_R [1, ..., q-1]$,

   1.3. $(k_1, k_2) = H(y_b^x \bmod p)$,

   1.4. Get current time-stamp TS,

   1.5. $c = E_{k_1}(key, TS)$,

   1.6. $r = KH_{k_2}(key, TS, etc)$,

   1.7. $s = x/(r + x_a) \bmod q$,

   1.8. Alice sends (c, r, s) to Bob.

2. *Bob:* Bob verifies (c, r, s) with unsigncryption.

   2.1. $(\mathrm{k}_1, k_2) = H((y_a \cdot \mathrm{g}^r)^{s \cdot x_b} \bmod p)$,

   2.2. $(\mathrm{key,TS}) = \mathrm{D}_{k_1}(c)$,

   2.3. $\mathrm{KH}_{k_2}(key, TS, etc) == r$ then message is valid else reject.

3. *Bob:* Bob generates a TS and signcrypt TS together with session key.

   3.1. $key^* \in_R \{0,1\}^{l_k}$,

3.2. $x^* \in_R [1, ..., q-1]$,

3.3. $(k_1^*, k_2^*) = H(y_a^{x^*} \bmod p)$,

3.4. Get current time-stamp TS^*,

3.5. $c^* = E_{k_1^*}(key^*, TS^*)$,

3.6. $r^* = KH_{k_2^*}(key^*, TS^*, key, etc)$,

3.7. $s^* = x^*/(r^* + x_b) \bmod q$,

3.8. Alice sends (c^*, r^*, s^*) to Bob.

4. *Alice:* Alice verifies (c^*, r^*, s^*) with unsigncryption. Then optionally, Alice sends a tag to Bob for secondary verification.

4.1. $(\mathrm{k}_1^*, k_2^*) = H((y_b \cdot \mathrm{g}^{r^*})^{s^* \cdot x_a} \bmod p)$,

4.2. $(key^*, TS^*) = D_{k_1^*}(c^*)$,

4.3. $KH_{k_2^*}(key^*, TS^*, key, etc) == r^*$ then message is valid else reject.

4.4. $K = key \oplus key^*$,

4.5. $tag = MAC_K(TS)$,

5. *Bob:* $K = key \oplus key^*$, and Bob verifies tag so that $tag == MAC_K(TS)$.

In HIMUTSIS, we preferred DKEUTS protocol. Usage of DKUTS protocol in military MANET, advantages and performance comparison is given in Chapter 7.

Another signcryption based technique that we have used in our security mechanisms is the multi-recipient signcryption scheme. We have utilized multi-recipient signcryption scheme in NAMEPS. Details and performance comparisons are given in Chapter 6.

We give description of multi-recipient signcryption scheme below [21]:

D. Signcryption by Alice the Sender for Multi-recipients

Alice sends message M to t recipients $R_1, ..., R_t$, public key of R_i is y_i for all $1 \leq i \leq t$, q and p and x_a is the private key of Alice.

1. Generate a random encryption key k', calculate $h = KH_{k'}(M)$, and encrypt M by $c = E_{k'}(M, h)$.

2. Create a signcrypted text of k' for each recipient $i = 1, ..., t$:

2.1. Generate a random number v_i from $[1, ..., q-1]$ and calculate $k'_i = H(y_i^{v_i} \bmod p)$. Split k'_i into $k'_{i,1}$ and $k'_{i,2}$ of appropriate length.

2.2. $c_i = E_{k'_{i,1}}(k')$,

2.3. $r_i = KH_{k'_{i,2}}(M, h)$,

2.4. $s_i = v_i/(r_i + x_a) \bmod q$.

3. Alice broadcasts to all the recipients $(c, c_1, r_1, s_1, ..., c_t, r_t, s_t)$.

E. Unsigncryption by Alice Each Recipients

Each recipients receive Alice $(c, c_1, r_1, s_1, ..., c_t, r_t, s_t)$. Also, each recipients R_i's private key x_i where $1 \leq i \leq t$, public key of Alice y_a, g, q and p.

1. Find out (c, c_i, r_i, s_i) in $(c, c_1, r_1, s_1, ..., c_t, r_t, s_t)$.

2. $k'_i = H((y_a \cdot g^{r_i})^{s_i \cdot x_i} \bmod p)$. Split k'_i into $k'_{i,1}$ and $k'_{i,2}$ of appropriate length.

3. $k' = D_{k'_{i,1}}(c_i)$,

4. $w = D_{k'}(c)$. Split w into M and h.

5. If h can be recovered from $KH_{k'}(M)$ and r_i recovered from $KH_{k'_{i,2}}(w)$.

6. If $((h == KH_{k'}(M))$ and $(r_i == KH_{k'_{i,2}}(w))$ then R_i accept M as a valid message else reject.

4. A NEW SATELLITE MULTICAST SECURITY MECHANISM BASED ON ELLIPTIC CURVE SIGNATURES

In this section, we present the TTPVSS (Two-Tier Pintsov-Vanstone Signature Scheme), which is a new satellite multicast security mechanism based on ECPVSS (Elliptic Curve Pintsov-Vanstone Signature Scheme) [40]. TTPVSS provides a basis and a generic structure for our other security mechanisms for SSMS. Notice that, TTPVSS utilizes some principles of [1]. TTPVSS is especially designed for very large satellite multicast systems having highly dynamic member join-leave characteristic. We design two independent key distribution tiered structure that has many advantages over classical satellite multicast systems. Our mechanism significantly reduces rekeying workload of the satellite that is the most resource limited part of the satellite multicast system. Also, number of keys that are stored on the satellite is reduced. As a novel approach, we use ECPVSS, which provides major cryptographic goals together while it significantly reduces bandwidth consumption, for secure key transmission. As a result, our security mechanism can handle very large multicast system securely and effectively while providing many advantages when compared to some other security mechanisms. In the following sections, we give details of TTPVSS.

4.1. Introduction to TTPVSS

Secure satellite multicast systems (SSMS) have critical importance in today's communication systems. Many real time applications such as military command and control, secure audio-visual data multicast, pay TV and file distribution applications need secure, reliable and high performance satellite multicast systems. However, providing security and effectively managing cryptographic keys in SSMS are challenging problems. Problems become much severe especially for SSMS, which have very large number of members and dynamic member join-leave characteristics. Satellite multicast systems are more vulnerable to security attacks. Eavesdropping and active intrusion

are much easier then terrestrial fixed networks. Also, SSMS are resource limited especially for power and bandwidth consumptions. One of the most important issues is that, in order to provide forward and backward security, whenever a member join-leave event occurs, group key must be updated (rekeying). Rekeying causes massive workload and significant performance problems especially for very large and dynamic SSMS [39].

In order to offer an efficient solution to aforementioned problems, we proposed TTPVSS security mechanism. Our security mechanism is especially designed for satellite multicast systems having very large number of members and high member join-leave frequency. TTPVSS consists of two new approaches and uses combined methods including new concepts for security mechanism designs and application suit cryptographic methods. Our security mechanism 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 forward and backward security requirement in SSMS. TTPVSS uses two independent key distribution tiers for solving this frequently rekeying problem. First tier consists of satellite-terrestrial units (TUs) and second tier consists of TU-members. Both tier uses LKH key distribution protocol. Using two independent key distribution tiers, effect of the modification is encapsulated on only its local group. Using independency of tiers 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 structural performance gains, our mechanism uses appropriate cryptographic algorithms that make two independent encryption tiers feasible and secure. In this mechanism, we use ECPVSS (Elliptic Curve Pintsov-Vanstone Signature Scheme) [40], [24] that satisfies many properties of ECDSA like authentication, integrity and unforgeability. However, ECPVSS is a message recovery type signature that is especially suitable for bandwidth constraint environment [40] and has advantages when compared to classical signature schemes. Our security mechanism 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.

4.2. Properties and Description of ECPVSS

In traditional SSMS approach, for cryptographic methods, generally, DLP based DH, public key cryptography algorithms such as RSA [41], ElGamal [27] and ECDH, which is extension of the DH in EC, are used [42]. However, these approaches do not provide critical cryptographic goals together, that are confidentiality, authentication, integrity and unforgeability. Especially, group based DH [43] and ECDH approach are vulnerable "man-in-the-middle attack" [1]. Classical digital signatures such as DSA and ECDSA provide these properties but causes bandwidth overheads. Note that, ECC based cryptographic methods have important advantages for both computational complexity and key storage and are preferred for wireless networks [44]. For bulk data multicast, symmetric key cryptography is used. Block ciphers in appropriate mode such as AES [45], DES [46] or stream ciphers can be used.

ECPVSS is a message recovery (MR) type signature scheme based on ECC. ECPVSS has many advantages for short messages when compared to the signature scheme with appendix [3] and some other MR type signature schemes. ECPVSS has been proposed in [40] 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 then classical digital signature algorithms.

Formal security proofs for ECPVSS is given in [48] covering ROM (Random Oracle Model). Also, some concrete examples are given for size of the messages used in DPM applications. Also, analysis, proofs and techniques for MR type signatures and ECPVSS can be found in [49]. Note that, ECPVSS has been standardized by IEEE in [47]. 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 nr in the group of points on elliptic curve $E(\mathbb{F}_q)$ over finite field $\mathbb{F}_q$ and number of points on the curve is divisible by nr. Then following notations are used:

γ : A point on the curve and is used as implicit certificate.

I_s: Identity of the signer.

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 the plaintext part of the data. In the description of ECPVSS, we directly use Q and do not show how it is generated from I_s, γ and 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 rk where $rk < n$.

3. $R = rk \cdot G$, R is a point on the curve.

4. Derive a symmetric key $\mathcal{K}$ from R using key derivation function $\mathcal{K} = KDF(R)$.

5. Transform the C using bijective transformation Tr parameterized by $\mathcal{K}$.

This transformation destroys the algebraic structure of C. Tr may be a symmetric encryption algorithm such as AES, DES or simply XOR operation: $u = Tr_R(C)$. Confidentiality of R is protected by intractability of the ECDLP and randomness of the value rk.

6. $d = H(u||I_s||V)$.

7. $b = a \cdot d + k \bmod n$.

8. Pair (b, u) is the signature pair which is used for verification. Pair (b, u) and plaintext data part V are sent to the verifier.

Signature Verification:

1. Public key of the signer Q is authentically obtained by verifier.

2. $d = H(u||I_s||V)$.

3. $U = b \cdot G - d \cdot Q$. Use KDF if it is necessary.

4. $X' = Tr_U^{-1}(u)$. Recover the confidentially protected part of the data.

5. Check redundancy of X' and if X' has required redundancy then declare $X' == C$ and accept the signature is valid.

4.3. Design Properties and Principles of TTPVSS

We present two major contribution of TTPVSS in the following sections. Section 4.3.1 gives contribution of TTPVSS for structural design aspect and section 4.3.2 gives contribution of TTPVSS for cryptographic technique aspect.

4.3.1. Contribution for Structural Design

Most important performance gain is obtained from structure design aspect of the SSMS. TTPVSS uses two independent key distribution tiers that provide significant performance gain especially for 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 operations and number of members that are affected by rekeying operations.

Taking into consideration these problems, we design two independent LKH tiers for SSMSs. In first tier (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 tier (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 do not affect from modification. This approach significantly reduces the workload of the satellite. Figure 4.1 present structural design of TTPVSS. Detailed analysis of TTPVSS is given in section 4.5.

LKH protocol is selected to use in tiers because LKH can handle moderately large and dynamic multicast groups successfully. Note that, in our mechanism, each TU manages moderately large groups due to independency principle.

4.3.2. Contribution for Cryptographic Method Aspect to the Multicast Security Mechanisms

In multicast security mechanisms, cryptographic methods that are used to transmit keys have critical importance. Even if key management protocol and structural design minimizes rekeying workload, if appropriate cryptographic methods are not used, then system is overloaded due to cryptographic processes. This situation especially must be taken into consideration for tiered structure like our security mechanism.

Classical key exchange methods are prevalently used but naive implementation of these protocols causes security problems. Notice that, to achieve 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 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 for use in satellite multicast security mechanisms. In group key update, only small symmetric keys are transmitted having 128 or 256 bits to the destination. Major idea behind of our choice is that an efficient MR type signature is an 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 4.2. As far as our concern, ECPVSS has not been used to transmit key update processes in satellite multicast security mechanism. Details of the advantages of the using ECPVSS are given in section 4.5.

4.4. Details of Our Security Mechanism

We give details of our security mechanism. Following notations are used:

ICI : Implicit Certificate Information.

EC_Keygen: Elliptic curve key generator that generates and validates ECC 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 belongs to a local TU member group.

4.4.1. Satellite-TU Tier

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 achieves major cryptographic goals for satellite-TU tier. GK is used for symmetric encryption of group key vectors z_i. TUs use group key vectors z_i to realize data multicast to the members. Notice 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.

$Q_i = EC_Keygen(\gamma_i, I_{s_i}, a_i)$, $ICI_i \leftarrow$ (Q_i and other required parameters for ECPVSS steps) where $1 \leq i \leq l$.

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.

$$GK = CPRNG(), C = GK, V_i = ICI_i,$$

$$(u_i, b_i) = ECPVSS_Sing(V_i, C),$$

$$TU_i \leftarrow (V_i, u_i, b_i), \ 1 \leq i \leq l.$$

3. Satellite generates group key vectors for z_i for $1 \leq i \leq l$. Each group key vector z_i is assigned to a TU. Elements of group key vector z_i are $z_{i,j}$ where $1 \leq i \leq l$ and $1 \leq j \leq n_s$. $z_{i,j}$ 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. Notice 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.

$$z_{i,j} = CPRNG(l, n_s), \acute{z_{i,j}} = E_{GK}(z_{i,j}),$$

$$TU_i \leftarrow \acute{z_{i,j}}, 1 \leq i \leq l, 1 \leq j \leq n_s.$$

4. Satellite realize data multicast using GK.

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

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

4.4.2. TU-Member Tier

In this tier, 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 the 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. $(V_i, u_i, b_i, M') \leftarrow Satellite$.

2. $C_i = ECPVSS_Unsign(V_i, u_i, b_i)$, $GK = C_i$ 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 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.

$Q_i' = EC_Keygen(\gamma_i', I_{s_i}', a_i')$, $ICI_i' \leftarrow$ (Q_i and other required parameters) where $1 \leq j \leq n_s$.

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

$C_i' = z_{i,j}$, $V_i' = ICI_i'$, $1 \leq i, j \leq n_s$.

$(u_i', b_i') = ECPVSS_Sign(V_i', C_i')$,

$M' = E_{z_{i,l}}(M)$, $Member_i \leftarrow (V_i', u_i', b_i', M')$.

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.

$(V_i', u_i', b_i', M') \leftarrow Member_i$, $C_i' = ECPVSS_Unsign(V_i', u_i', b')$, $z_{i,j} = C_i'$ 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.

4.5. Performance Comparison and Results

We give comparison of our security mechanism to some other well-known security mechanism for two major aspects. Firstly, we give advantages of TTPVSS resulting from its structural design. Then, we present advantages of TTPVSS obtained from its novel cryptographic method approach.

4.5.1. Advantages and Performance Comparison for Structure and Design Aspects

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

Table 4.1 shows workload comparison of our security mechanism 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 chapter 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 security mechanism, 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 security mechanism 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 mechanism 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 security mechanism, rekeying workload of the satellite is $(2 \log_2 500 \approx 20)$ that is much smaller than Flat and LKH ($m' = 1$). 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 $\mathrm{r}_l = r/l$. 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 an another important parameter. In both Flat and LKH, unique keys are stored in satellite for each member and storage load is N. In our security mechanism, 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 security mechanism, it is $\log_k(n_l) \approx 11$.

4.5.2. Advantages of Selected Cryptographic Methods

As a novel approach, we use a message recovery type algorithm ECPVSS in our satellite multicast security mechanism. Main goal of cryptographic routines in multicast security mechanisms 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 4.2 shows comparison of ECPVSS to other well-known cryptographic methods that are also frequently used in SSMS. Possible message length of ECPVSS signature and message overheads are given in 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 showed. Note that, in Elgamal encryption, ciphertext is doubled [2]. 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 [40]. 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 mechanism, 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 three times better then nears competitive for bandwidth consumption. Table 4.2 summaries these results.

4.6. Conclusions

In TTPVSS, we propose a new satellite multicast security mechanism that have many advantages over classical multicast security systems. Our security mechanism uses two independent key distribution tiers (satellite-TU and TU-Members tiers) that significantly reduce the rekeying workload of the satellite, which is the most resource limited component of the satellite multicast system. Also, number of keys that are stored in the satellite is reduced. This structure encapsulates rekeying operations in local member groups in TU-Member tier 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 for very large and dynamic multicast systems. In addition to this, as a novel approach, we propose to use ECPVSS for major cryptographic method in satellite multicast systems. ECPVSS is a MR type signature scheme based ECC and specifically designed for bandwidth constraint environments. We use ECPVSS for secure group key and seed transmission that provides at least three times bandwidth advantages for best competitive method. Moreover, ECPVSS provides major cryptographic goals efficiently and together while many methods can partially provide these properties. As a result, TTPVSS can be used securely and effectively to manage very large and dynamic satellite multicast groups and have many advantages to the some well-known mechanisms.

Table 4.1. Workload comparison of TTPVSS to Flat and LKH

For very large systems, $l > 500, k = 2, r > 10^5$ and $N > 10^6$, $n_l = N/l$			
	Rekeying load over satellite	# keys stored in satellite	# keys stored in members
Flat	$N \cdot r$	N	l
LKH	$(klog_kN) \cdot r$	N	log_kN
Our Mechanism	$(k \log_k l)/m$	$2 \cdot l + 1$	$\log_k(n_l)$

Table 4.2. Advantages and properties of ECPVSS against its widely used alternatives

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								

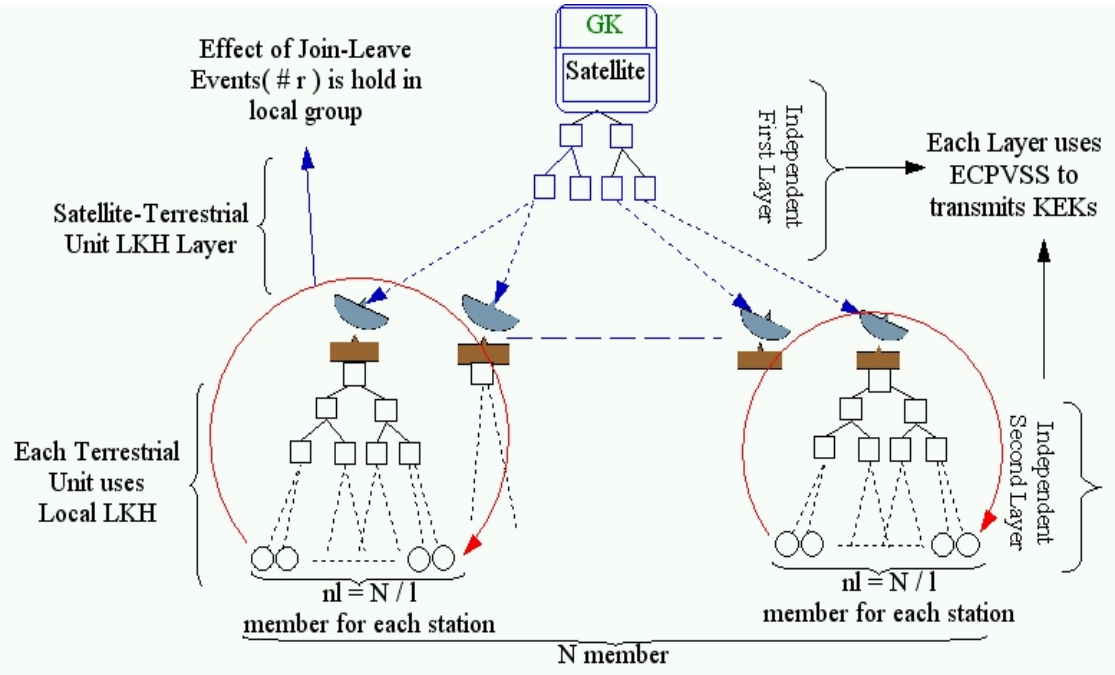


Figure 4.1. Structure of TTPVSS

5. THREE-TIER SATELLITE MULTICAST SECURITY MECHANISM BASED ON ECMQV AND IMC METHODS

In this section, we present a new three-tier satellite multicast security mechanism [13], which is based on ECMQV (Elliptic Curve Menezes-Qu-Vanstone) and IMC (Improved Merkle Cryptosystem). Our three-tier ECMQV based mechanism utilizes some principles of TTPVSS [16] but it is completely different from TTPVSS for its structural design and cryptographic techniques aspects. In this mechanism, we make contribution to the satellite multicast security mechanisms in two major points. These are structure design and cryptographic methods aspects. Similar to TTPVSS, our three-tier satellite multicast security mechanism is specifically designed for multicast systems having very large number of members and highly dynamic member join leave characteristic. It minimizes the rekeying workload of satellite tiers and shows high performance for many criteria using three independent key distribution tiers. In addition, our security mechanism uses a different cryptographic method for each tier and achieves major cryptographic goals together that are not provided in the implementations of some other protocols. Using ECMQV and especially IMC based methods in satellite multicast security mechanisms is a novel approach and has many advantages. Details of our three-tier satellite security mechanism are given below.

5.1. Introduction to Our Three-Tier Satellite Multicast Security Mechanism

As we have mentioned in chapter 4, satellite multicast applications gain significant importance in today's applications such as military command and control, secure audio-visual data multicast, pay TV and file distribution applications. However, when the number of members in multicast group increases significantly, then providing security in SSMS becomes much harder. Especially for SSMSs, which is designed to cover large areas of the world, requires more efficient and integrated approach to provide high performance and security in their applications than traditional SSMS approaches.

Providing forward and backward security together with major cryptographic goals without creating significant performance problem is major purpose of SSMSs.

In this study, taking into consideration these problems, we propose a new three-tier satellite multicast security mechanism based on ECMQV (Elliptic Curve Menezes-Qu-Vanstone) [25], IMC (Improved Merkle Cryptosystem) [23] and ECPVSS (Elliptic Curve Pintsov-Vanstone Signature Scheme) [47] cryptographic protocols. Our security mechanism can be applied to very large multicast groups with 10^6 members or more without creating performance and security problems. Our mechanism is also suitable for highly dynamic multicast groups having mobile members.

We use three independent key distribution tiers to provide scalability and modularity to handle very large multicast systems. The satellite tiers of our mechanism consist of a GEO satellite tier as the general manager of system and a LEO-MEO satellite tier for bulk data multicast and key distribution purposes. The third tier of our mechanism includes terrestrial units (TU), non-trusted servers (NTS) and members. Each tier uses appropriate cryptographic algorithms and key establishment protocols with their alternatives. Using this tiered structure provides many benefits such as decentralization of the multicast and key distribution workloads as well as the centralization of security control.

We use alternative cryptographic methods in our security mechanism such as ECMQV, IMC and ECPVSS to the naive key exchange and classical RSA and DLP (Discrete Logarithm Problem) based signature approach [42]. The reason is that ECMQV provides efficient and secure collaborative (fair) key exchange between tiers when compared to the classical approaches. IMC and IMC based methods are novel approaches and have not been used before in satellite multicast security mechanisms. ECPVSS provides bandwidth and security advantages in the third tier.

Our security mechanism provides significant advantages for the rekeying workload of the satellite tier, which is one of the most important parameters for satellite multicast systems. It also provides advantages to reduce the number of keys which are stored in

satellites and TUs. Apart from these, central security management with decentralized multicast workload is an important advantage.

5.2. Cryptographic Techniques Used in Our Three-Tier Satellite Security Mechanism

In our security mechanism, we essentially use ECC based techniques. ECC has many advantages over factorization based [41] and classical DLP [27] based approaches for both computational and key bit length aspect [44]. Application of ECPVSS algorithm to the satellite multicast security, TTPVSS, is given in chapter 4. In our three-tier satellite security mechanism, we use ECMQV key agreement protocol [50] as a major cryptographic primitive. Also, as a novel approach, we show that it is possible to use IMC or IMC based methods. We give brief descriptions of these protocols and present their advantages.

5.2.1. The ECMQV Protocol

The ECMQV protocol is an authenticated key agreement protocol, proposed by Law, Menezes, Qu, Solinas and Vanstone [25], based on the standard authenticated Diffie-Hellmann (DH) key agreement protocol on EC. The ECMQV protocol provides known-key security (KK-S), forward secrecy (FS) and key-compromise impersonation resilience (KCI-R) under the assumption of the intractability of ECDLP. In KK-S, even if sessions are revealed to an adversary, each execution of the key agreement protocol must generate unique and matching session key. In FS, even if long term private keys are revealed to an adversary, the secrecy of previously established session keys should not be affected. In KCI-R, even if one of the instances of communication is corrupted due to private key loss, then an adversary can not masquerade Alice as another principal [51]. Notice that these properties can not be achieved by classical DH based approaches. The ECMQV protocol is also standardized by IEEE P1363. Details can be found in [42]. Moreover, some improvement for MQV, HMQV (Hashed MQV) is also proposed in [26]. It uses hash functions and challenge-based signatures. The study in [52] gives a different approach to HMQV.

The ECMQV protocol is given in Table 5.1. F is a finite field, E is an elliptic curve, $\#E(F)$ is the number of points on elliptic curve and q is the prime divisor of $\#E(F)$ and also is the order of the curve point G on subgroup of order q. Let x the binary representation of the first coordinate of Q. Let $\overline{Q}$ be defined as $\overline{Q} = x \bmod 2^{\lfloor f/2 \rfloor} + 2^{\lfloor f/2 \rfloor}$ ($Q \leftrightarrow R$ and $\overline{Q} \leftrightarrow \overline{R}$ in Table 5.1).

Table 5.1. The ECMQV Key Establishment Protocol

Alice		Bob
1. Alice generates a static private key $w_a \in \{1, \cdots, q-1\}$, and compute $W_a = w_a \cdot G$. Publish static public W_a	$\Leftrightarrow$	1. Bob generates a static private key $w_b \in \{1, \cdots, q-1\}$ and compute $W_b = w_b \cdot G$. Publish static public W_b
2. Alice generates the ephemeral private key $r_a \in \{1, \cdots, q-1\}$, compute the corresponding ephemeral public key $R_a = r_a \cdot G$ and send this value to Bob	$\Leftrightarrow$	2.Bob generates the ephemeral private key $r_b \in \{1, \cdots, q-1\}$, compute the corresponding ephemeral public key $R_b = r_b \cdot G$ and send this value to Alice
3. Alice compute $s_a = (r_a + \overline{R}_a w_a) \bmod q$ and $\mathrm{R_b} + \overline{R}_b W_b = s_b \cdot G$		3.Bob compute $s_b = (r_b + \overline{R}_b w_b) \bmod q$ and $R_a + \overline{R}_a W_a = s_a \cdot G$
4.Alice compute $Z = \frac{\#E(F)}{q} \cdot s_a \cdot (R_b + \overline{R}_b W_b)$		4. Bob compute $Z = \frac{\#E(F)}{q} \cdot s_b \cdot (R_a + \overline{R}_a W_a)$
5. Both Alice and Bob generates shared secret key Z $Z = (\#E(F)/q) \cdot s_a \cdot s_b \cdot G$		

5.2.2. The IMC and IMC Based Methods

In chapter 3, we have mentioned some of the well-known public key cryptosystems. Generally, key agreement methods including digital signatures are based on these public key cryptosystems. However, the first cryptosystem, which provides a solution to the secure communication problem over insecure channels without pre-established secrets, is the Merkle Cryptosystem (MC) [54]. In this study, as an alternative key

establishment method, in addition to ECMQV and ECPVSS, we offer to use IMC, which has been proposed in [23]. IMC increases the security of the original MC and its variants (VMC) in [55]. In IMC, cryptographic hash functions and a new puzzle structure are used together in order to increase the security of MC and VMC. The key agreement value, which is sent as clear text in VMC, is hidden using a cryptographic hash function in IMC. Also, in order to increase the security of the key agreement value, auxiliary keys are used. In addition to IMC, we work on STAKE protocol achieving major cryptographic goals such as confidentiality, authentication, integrity and unforgeability together. STAKE is based on IMC and signcryption type key establishment schemes. Most important property of STAKE is that it only relies on symmetric cryptosystems and cryptographic hash functions while achieving major cryptographic goals. IMC based approaches do not need any trusted third party even if it relies on only symmetric cryptosystems and cryptographic hash functions.

IMC and IMC based methods are novel approaches having these properties and we propose them as alternative methods to use in key establishment steps in a satellite multicast system. These methods are especially suitable for applications that require high security and prefer to use a symmetric cryptosystem. IMC can provide approximately 3000 bit RSA equivalent and 2500 bit sub group DLP equivalent security. However, as an inherent property of MC, IMC and IMC based approach (STAKE) have non-neglible storage requirements. For this reason, if enough storage is not available then implementing only ECMQV can be more appropriate. However, IMC and IMC based approach are quite applicable for today's standard hardware systems under usual conditions. We give details of IMC in chapter 8.

5.3. Structure and Design Properties of Our Three-Tier Security Mechanism

5.3.1. Major Design Properties

We design a new satellite multicast security mechanism having three independent LKH tier. Each tier uses appropriate cryptographic algorithms and key establishment

protocols together with their alternatives. These cryptographic methods and mechanisms provide solutions for key establishment among tiers. Each mechanism is selected taking the properties of each tier into consideration.

Our security mechanism utilizes the major design principles of the tiered structure, which are the independency and modularity principles. Hybrid key distribution mechanisms use the divide-and-conquer approach to tackle high rekeying and cryptographic workload of multicast security systems. As we mention in chapter 2, to handle systems having very large number of members and high member join-leave activity rate, a combination of key hierarchy and group based hierarchy approaches has been proposed such as [12]. These mechanisms are designed for general multicast security systems. Applying the independency and modularity principles specifically on satellite multicast systems; TTPVSS provides advantages in terms of both computational effort and rekeying workload.

We use the design principle of TTPVSS in our three-tier approach while improving it in many aspects. We utilize existing tiered structure of satellite networks, which is not studied in TTPVSS. LEO, MEO and GEO satellites, already having a hierarchical structure, can be used to design more efficient key distribution mechanisms. Using different properties of these satellite tiers, the overall system performance can be increased and the workload of the individual satellites can be reduced.

5.3.2. Structure of Our Security Mechanism

Our security mechanism consists of three tiers: GEO satellite tier, LEO-MEO satellite tier and terrestrial unit (TU)-member tier. Communication among tiers is realized in hierarchical manner taking modularity principle into consideration . In each tier, whenever a member join-leave event occurs, LKH is applied to the related local group. The GEO satellite applies LKH to its LEO-MEO satellite tier. Each LEO-MEO satellite has its own TU group and applies LKH this local TU group. Similarly, each TU has its own member group and applies LKH to this local member group. LKH is preferred like TTPVSS since it has computational and storage advantages for

multicast systems as described in chapter 2 and 4.

NTS are used to store public key parameters for the ECMQV and IMC protocols. Public keys are transmitted to NTS only once and whenever they are needed, they are obtained from NTS. Consequently, the satellite tier is not affected by future key agreements that will be realized with the same public keys.

The detailed description of our security mechanism for each tier is given in the next section. Now, we briefly explain properties and responsibilities of each tier.

5.3.2.1. GEO Satellite Tier. GEO satellite tier is responsible for general key management of the overall multicast system. The group keys, which are generated by the GEO satellite, are transmitted to each tier in encrypted form. In this mechanism, group keys of the lower tier are known by only the upper tiers and keys are only determined by the GEO satellite. This way, the GEO satellite can always control and manage the overall multicast system.

Firstly, the GEO satellite(s) realizes ECMQV key exchange to transmit group keys and seeds to the LEO-MEO satellite tier. Secondly, it generates and transmits group key seeds, which will be used between the LEO-MEO satellite and TUs, to the LEO-MEO satellite tier. Thirdly, it generates and transmits group key seeds that will be used between TUs and members to realize a secure communication. These session keys and seeds should be generated by a CPRNG (Cryptographic PRNG) like Blum-Blum-Shub [53] because they will be used for long term security as a principle of batch keying. Also, as an option, if needed, the GEO satellite may involve data multicast.

5.3.2.2. LEO-MEO Satellite Tier. This tier is mainly responsible for bulk data multicast to TUs and distributing group keys to the TUs and NTS. LEO-MEO satellites obtain shared secret keys from the GEO satellite. Each LEO-MEO satellite uses these shared secret keys to obtain the related group key to communicate with the upper tier. The group key seeds are used to generate group keys. Then, they distribute the

group keys and session key seeds to the TU using either ECMQV or IMC protocol also involving NTS if necessary.

5.3.2.3. TU-Member Tier. In this tier, TUs are responsible for decrypting the data coming from the LEO-MEO satellite tier and multicasting it after encrypting it with required group keys. TUs obtain and use group keys to realize secure communication with the LEO-MEO satellite tier using either ECMQV or IMC protocol also involving NTS if necessary. TUs use seeds to generate group keys that will be used for secure bulk data multicast. Figure 5.1 shows the structure of our security mechanism.

5.4. Details of Our Three-Tier Security Mechanism

In each tier, there is a hierarchical key exchange and transmission from top to down. The GEO satellite is responsible for generating and distributing seed values $s\alpha_i$ and $s\beta_i$. These seed values are used to generate group key vectors α_i and β_i. Each $s\alpha_i$ seed value is assigned to a LEO-MEO satellite by the GEO satellite. Each LEO-MEO satellite generates group key vector α_i using seed $s\alpha_i$. Elements of group key vector α_i are $\alpha_{i,j}$ where $j \geq l$. So, $\alpha_{i,j}$denotes j^{th} group key which is used by i^{th} satellite in LEO-MEO satellite tier. Each satellite realizes bulk data multicast using these group keys $\alpha_{i,j}$ to their local TU groups and also transmits seed values $s\beta_i$ to the related TUs.

Each TU realizes the ECMQV key exchange with the upper tier to obtain and transmit the required keys. Each $s\beta_i$ seed value is assigned to a TU. Each TU generates the group key vector β_i using seed $s\beta_i$. Elements of group key vector β_i are $\beta_{i,j}$ where $j \geq n_l$. $\beta_{i,j}$ denotes j^{th} group key for i^{th} TU in TU-Member tier. TUs use group keys $\alpha_{i,j}$ to decrypt multicast data and seed values $s\beta_i$. Using ECPVSS, different from upper tiers, TUs transmit $\beta_{i,j}$ keys to members and realize bulk data multicast using these group keys $\alpha_{i,j}$. Members, using ECPVSS, obtain group keys and decrypted bulk multicast data securely. ECPVSS is preferred in this tier because "fair key exchange" may not be preferred in TU-Member hierarchy. Thus, only key transport is realized

by ECPVSS like TTPVSS.

Seed vectors $s\alpha$ and $s\beta$ are used for batch keying: Thus, instead of sending group key vector elements in α_i and β_i one by one, only their seed values are transmitted. LEO-MEO satellites generate group keys using these seeds and important bandwidth and rekeying cost advantages are gained. Details of each step for tiers are given in the following part, where:

n_s : Number of satellites in LEO-MEO satellite tier.

l : Number of TUs in TU-Member tier.

N : Total number of members in multicast system.

$n_l \approx N/l$:Average number of members in one TU local group.

$ECMQVKG$: Static Key Generation for ECMQV protocol.

A. GEO Satellite Tier

The GEO satellite generates required group keys and seeds for overall system. Also, if it is necessary, the GEO satellite may realize bulk data multicast.

A.1) GEO satellite generates the group key seed:

$s\alpha_i = CPRNG(\geq n_s)$, $s\beta_i = CPRNG(\geq l)$.

A.2) GEO satellite generates static public-private key pairs from EC curve E and validates them. ECMQV key exchange is done using public keys of lower tier W_{b_i} and required private keys. Using these, GEO satellite has shared secret keys (unique keys) with each satellite at the lower tier:

$(W_{a_i}, w_{a_i}) = ECMQVKG(E, n_s)$, $NTS_i \leftarrow W_{a_i}$ and $W_{b_i} \leftarrow NTS_i$,

$Z_i = ECMQV(W_{b_s}, w_{a_i}$, required private keys) unique keys for each satellite in LEO-MEO satellite tier.

A.3) GEO satellite generates and distributes group key GK to the all satellites in LEO-MEO satellite tier using secret shared key Z_i. This group key is used to transmit group key seeds and bulk data multicast if necessary:

$GK_i^{'} = E_{Z_i}(GK)$, for each satellite in LEO-MEO tier.

A.4) GEO satellite multicast seed vectors $s\alpha$ and $s\beta$ using GK. Vector $s\alpha$ is used to generate group keys $\alpha_{i,j}$ that LEO-MEO satellites will use to communicate with TUs. Vector $s\beta$ is used in TU-Member tiers. Optionally, bulk data multicast can be realized.

A.5) During each member join-leave, LKH protocol is applied to the LEO-MEO satellite tier using GK to update the required keys.

Optionally, If IMC is used; GEO satellite(s) generates seed vectors st, sx and transmits them to the LEO-MEO satellite tier. These seed vectors are used to generate private-public keys for IMC based cryptosystem:

$st_i^{'} = E_{GK}(st_i)$, $sx_i^{'} = E_{GK}(sx_i)$.

B. LEO-MEO Satellite Tier

LEO-MEO satellites, using satellite inter-networking, determines which TUs are managed by which satellite. Group key and group key seed distribution are done according to this agreement.

B.1) LEO-MEO satellites generate static public-private key pairs from EC curve

E and validate them. ECMQV key exchange is done using public keys of upper and lower tiers using required private keys. Z_i secret keys are shared with the GEO satellite and Z_i' secret keys are shared with TUs. NTSs are used to store and obtain public keys. Using Z_i, each satellite obtains group key GK:

$(W_{b_i}, w_{b_i}) = ECMQVKG(E, n_s)$, $NTS_i \leftarrow W_{b_i}$ and $W_{a_i} \leftarrow NTS_i$,

$Z_i = ECMQV(W_{a_i}, w_{b_i}$,required private keys).

$GK = D_{Z_i}(GK_i')$ realized for each LEO-MEO satellite.

$(W_{b_i}', w_{b_i}') = ECMQVKG(E, l)$, $NTS_i \leftarrow W_{b_i}'$ and $W_{t_i} \leftarrow NTS_i$,

$Z_i' = ECMQV(W_{t_i}, w_{b_i}'$,required private keys) realized for each TU.

B.2) Using GK, LEO-MEO satellites obtain group key seeds $s\alpha$ and $s\beta$. Each satellite uses its seed $s\alpha_i$ to generate vector α_i. Suppose that, i^{th} satellite uses group key $\alpha_{i,j}$ at the current state. After that, whenever a key update occurs, the i^{th} satellite uses next group key such that $\alpha_{i,j} \rightarrow \alpha_{i,j+1}$. This group key is used to manage the TU group which the satellite is responsible for.

$s\alpha_i = D_{GK}(s\alpha_i')$, $s\beta_i = D_{GK}(s\beta_i')$. If the GEO satellite sends message, $M = D_{GK}(M')$. Using $s\alpha_i$, sufficient number $\alpha_{i,j} = CPRNG(s\alpha_i, \geq l)$ group key is generated $(j \geq l)$.

B.3) Each LEO-MEO satellite sends group key to the TUs that are responsible for using shared secret key Z_i'. Also, seeds $s\beta_i$ are transmitted to the related TUs:

$\alpha_{i,j}' = E_{Z_i'}(\alpha_{i,j})$, $s\beta_i' = E_{Z_i'}(s\beta_i)$.

B.4) Bulk data multicast is done to the TUs using group keys: $M' = E_{\alpha_{i,j}}(M)$.

B.5) Each LEO-MEO satellite applies LKH its local TU group independently for each member join-leave event.

Optionally, If IMC is used, only MEO satellites obtain seed vectors st, sx and generate IMC secret key vector K using st. Then using K and sx, satellite generates public key vectors and transmits them to the NTSs. Only MEO satellites are involved in this process since they have higher computational and bandwidth resources.

C. TU-Member Tier

TUs are responsible for realizing the bulk data multicast to the members. Each TU belongs to a TU group, which is managed by a LEO or MEO satellite. Also, each TU has a large member group.

C.1) Like the upper tier, TUs generate static public-private key pairs from EC curve E and validate them. ECMQV key exchange is done using public keys of the upper tier using the required private keys. $Z_i{}'$ secret keys are shared with the LEO-MEO satellite tier. NTSs are used to store and obtain public keys. Note that TU sends group keys to the members using ECPVSS, which has advantages if collaborative (fair) key exchange is not needed. The following steps are performed for TU and members:

$(W_{t_i}, w_{t_i}) = ECMQVKG(E, l)$, $NTS_i \leftarrow W_{t_i}$ and $W_{b_i}{}' \leftarrow NTS_i$,

$Z_i{}' = ECMQV(W_{b_i}{}', w_{t_i}$,required private keys).

$\alpha_{i,j} = D_{Z_i{}'}(\alpha_{i,j}{}')$ and $s\beta_i = D_{Z_i{}'}(s\beta_i{}')$ realized by LEO-MEO satellite. Each TU generates group keys from seeds $s\beta_i$, $\beta_{i,j} = CPRNG(s\beta_i, \geq n_l)$, $j \geq n_l$. Each TU decrypts the bulk multicast data using group key from which they obtain LEO-MEO satellite. Then, TUs realize bulk data multicast to the members using group keys:

$M = D_{\alpha_{i,j}}(M')$ and $M' = E_{\beta_{i,j}}(M)$. Each TU transmits group key $\beta_{i,j}$ to the members by using $NTS_i \leftarrow$Public parameters for ECPVSS.

C.2) Members obtain group keys and decrypt bulk multicast data securely: Each member obtains group key $\beta_{i,j}$ from TU. (Public key parameters for ECPVSS)$\leftarrow NTS_i$, $\beta_{i,j} = ECPVSS_Unsign(\beta_{i,j}, PVSS_Parameters)$. Members decrypt multicast data $M = D_{\beta_{i,j}}(M')$.

5.5. Performance Comparison and Results

In this mechanism, two aspects of satellite multicast security mechanisms are analyzed and improved. Firstly, appropriate cryptographic methods are selected and integrated to our satellite multicast security mechanism. Secondly, structure and design properties of the satellite multicast security mechanisms are improved. We firstly focus on the performance gain and advantages of our security mechanism for its design and structure aspects. Then, we give advantages of cryptographic methods, which are used in our security mechanism when compared to classical approaches.

5.5.1. Performance Comparison of Design Aspects of Our Security Mechanisms

Table 5.2 shows performance comparison of our security mechanism to Flat, LKH and TTPVSS. The comparison is based on five major criteria: Rekeying workload for satellite tier and TU, number of keys stored in satellite tier, in TUs and members on the average. In table 5.2, average rekeying workload and number of keys in satellite tier (SL) are only applicable to our security mechanism for GEO and LEO-MEO SL. Typical parameter sizes can be considered as, $N \geq 10^6, r \approx 10^5, l = (500 - 1000), n_l = N/n_s \geq 2048, n_s \approx 100, m_2 > m_1, n_{tu} = l/n_s$.

The most important criterion is the average rekeying workload of the satellite tier. Since the most resource limited part of the satellite multicast system is the satellite tier, we aim to minimize rekeying workload of this part. For this criterion, among the compared security mechanisms, the most efficient one is our proposed security mechanism: In Flat and LKH, for each member join-leave event, the key update cost is N and $k \log_k N$ respectively. Notice that, the major parameter that determines the rekeying

workload of the system is the rekeying factor r, which is the total number of member join-leave events for a certain time period. In Flat and LKH, since the satellite directly controls all members, a rekeying occurs for each member join-leave event, according to the key update rule of the mechanism. Thus, total rekeying workload for Flat and LKH are $N{\cdot}r$ and $(k \log_k N){\cdot}r$ respectively. For the TTPVSS, satellite tier is not affected from member join-leave events due to independency of tiers principle. Thus, rekeying of workload of the satellite tier in our previous mechanism is $k \log_k l/m_1$ where l is the number of TUs in the related local TU group and m_1 is the batch keying factor. In our proposed security mechanism, rekeying workload of the LEO-MEO satellite determines the average rekeying workload for all satellite tiers. n_{tu} is the average number of TUs, which are controlled by a single LEO or MEO satellite. Thus, rekeying only occurs for a satellite if one of the TUs is down or violates the security policy. Notice that, these events occur rarely. Also, the batch keying factor becomes $m_2 > m_1$ since only seed values are transmitted. This makes possible to realize more batch keying. In addition to this, parameter m_2 reflects the advantages of lower packet loss and propagation delay values of LEO-MEO satellites when compared to our previous security mechanism. Thus, the rekeying factor of LEO-MEO satellite tier is $(k \log_k n_{tu})/m_2$. Notice that the independency principle is also valid for our proposed mechanism and rekeying factor r does not affect the satellite tier. The rekeying workload of the GEO satellite is small and neglible: $k \log_k n_s \leq 10$. As a result, since $n_{tu} << l$ and $m_2 > m_1$, our proposed security mechanism is more efficient than our previous security mechanism.

For average number of keys stored in satellite tier criterion, Flat and LKH require N unique keys for each member. In our previous security mechanism, the satellite only contacts with TUs and the storage requirement is l. In the proposed mechanism, in order to obtain performance gain for the rekeying workload, we slightly increase the number of keys stored in the satellite tier. Notice that the major parameter that determines average number of keys stored in the satellite tier is the storage requirement of the GEO satellite. In ECMQV protocol, key pairs $W_{a_i}, w_{a_i}, r_{a_i}, R_{a_i}$ Z_i and seed values $s\alpha_i$ and $s\beta_i$ are stored in GEO satellite. Thus, the average number of keys stored in the satellite tier is $6n_s + l + 1 \approx 2l$. The number of keys stored in the LEO or MEO satellite is small and neglible: $n_{tu} \leq l$.

Rekeying workload of TUs is only analyzed for previous and our proposed mechanism. For Flat and pure LKH, TUs are not specifically mentioned and are not considered in this comparison. Rekeying workload of the TU is the same for both our previous and proposed mechanisms, that is $\log_k n_l$. The reason is that, in both mechanism, TUs are only responsible for their own local member group for rekeying processes. Also, the number of keys that TU and a member stores are the same for both mechanisms and is $n_l + \log_k l$ and $\log_k l$, respectively. The number of keys that a member stores is $\log_k N$ in LKH and one in Flat (directly communicates with satellite).

5.5.2. Advantages of Cryptographic Methods Used in Our Proposed Security Mechanism

In our proposed security mechanism, ECPVSS, ECMQV and alternatively IMC algorithms are used. Table 5.3 shows a comparison of these algorithms and other prevalently used approaches in key exchange methods. DH-ECDH protocols are frequently used for key exchange. However, pure implementation of these protocols is insecure. Man-in-the-middle attack is the most well-known attack used against these protocols. Also, these protocols do not provide KK-S, FS and KCI-R security properties. Notice that for signature variants and ECPVSS; KK-S, FS and KCI-R are not compared because signature variants and ECPVSS are not key exchange protocols.

In table 5.3, taking into consideration the properties of algorithms, it is shown whether the algorithm achieves the mentioned property (authentication, integrity, unforgeability) or not. Also, for three criteria, bandwidth efficiency, computational effort and confidentiality, VL (very low), L (low), M (moderate), H (High) and VH (very high) levels are assigned. These assignments are done comparing algorithms with each other for their computational efforts, storage requirements and achieved security (equivalent bit length security). Bandwidth efficiency assignment is done according to the required packet bit length to realize a cryptographic method. Computational efforts are assigned according to the computational complexity of algorithms and various implementation considerations. Confidentiality comparison is done according to the cryptanalysis properties of algorithms. For different metrics, comparison and properties of

these mechanisms can be found in [25], [47], [44], [46], [37]. For instance, signature variants achieve authentication, integrity and unforgeability but they require transmission of signature together with the message. For small messages, this situation causes significant bandwidth consumption. ECPVSS is a message recovery type signature and solves this problem efficiently. IMC-STAKE also achieves major cryptographic primitives and provides high security. However, it is not appropriate for bandwidth constraint applications.

5.6. Conclusions

In this study, we propose a new satellite multicast security mechanism that provides many advantages compared to some well-known multicast security mechanisms and as well as our previous security mechanism. Using three independent LKH tiers provides significant performance gain for rekeying workload of the satellite tier and reduces the number of keys that are stored in satellites. Our security mechanism makes possible the centralization of security management and the decentralization of multicast workload at the same time. Moreover, using LEO-MEO satellite inter-networking, the delay problem is minimized and the workload of the individual satellite is reduced. Another benefit of our security mechanism is that it increases batch keying factor. Using ECMQV in the satellite tier provides "fair key exchange". It also achieves many cryptographic primitives, which can not be achieved by some other security mechanisms. IMC and IMC based methods are newly proposed and have not yet been used in satellite multicast security mechanisms providing some advantages. Also, advantages of ECPVSS in our previous security mechanism (TTPVSS) remain the same in the third tier of our mechanism. Advantages of our proposed mechanism to the Flat, LKH and TTPVSS can be observed in Table 5.2 and Table 5.3.

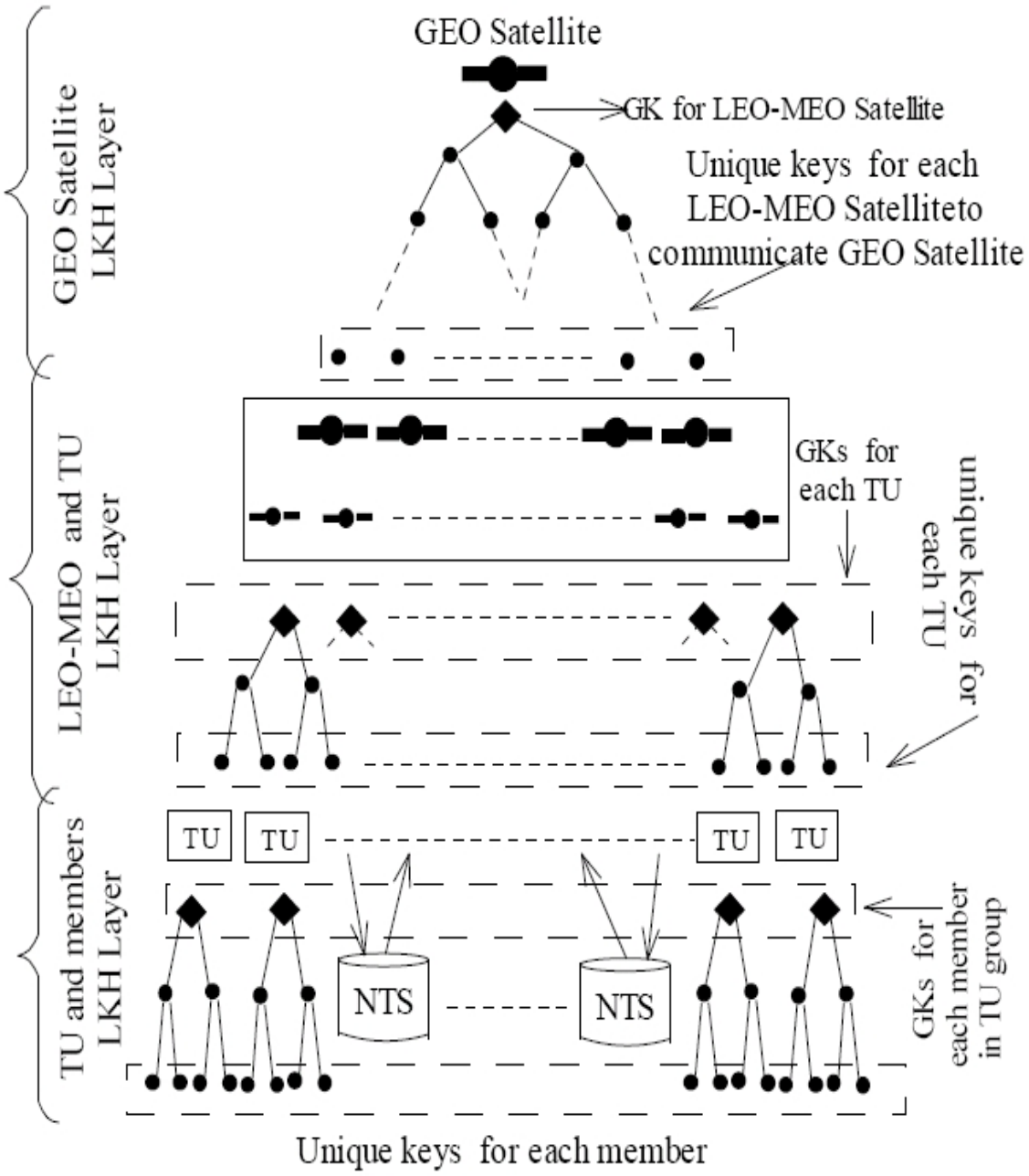


Figure 5.1. Structure of our satellite security mechanism

Table 5.2. Performance comparison of our mechanism to Flat, LKH and our previous mechanism (TTPVSS) is given for five major criteria

	Rekeying load over satellite	# keys stored in satellite	# keys stored in members
Flat	$N \cdot r$	N	l
LKH	$(klog_kN) \cdot r$	N	log_kN
TTPVSS	$(k \log_k l)/m_1$	l	$\log_k(n_l)$
Our Proposed Security Mechanism	$(k \log_k q)/m_2$	$6n_l + l + 1 \approx 2l$	
GEO	$k \log_k n_s \leq 10$	$6n_l + l + 1 \approx 2l$	
MEO-LEO	$(k \log_k q)/m_2$	$q < l$	

Table 5.3. Comparison of cryptographic protocols with regard to nine essential criteria. RSA-S denotes RSA Signatures and DSA-V denotes DSA Variants

	DH	ECDH	RSA-S & DSA-V	ECPVSS	STAKE	ECMQV
Authentication	No	No	Yes	Yes	Yes	Yes
Unforgeability	No	No	Yes	Yes	Yes	Yes
Integrity	No	No	Yes	Yes	Yes	Yes
KK-S	No	No	-	-	Yes	Yes
FS	No	No	-	-	Yes	Yes
KCI	No	No	-	-	Yes	Yes
BW Efficiency	M	H	L	VH	VL	H
Comp. Efficiency	M	H	M	VH	M	H
Confidentiality	H	H	H	H	VH	H

6. NAMEPS: N-tier sAtellite Multicast sEcurity Protocol (Mechanism) based on Signcryption schemes

In this chapter, we propose a new N-tier sAtellite Multicast sEcurity Protocol (Mechanism) based on multi-recipient Signcryption schemes (NAMEPS) [70]. NAMEPS utilizes some properties of TTPVSS and three-tier satellite multicast security mechanism [13] but improves them in many other points. NAMEPS is especially designed for very large and highly dynamic satellite multicast systems which require high security and reliability. Our N-tier structure significantly reduces workload of the satellite tiers especially for bandwidth consumption, computation resources and storage requirements. N-tier approach localizes effects of the rekeying operation (forward-backward security) and provides significant performance gain. Moreover, batch keying and ticketing mechanisms are used which additionally reduce workload of the satellite and terrestrial tiers. This batch keying mechanism is better than that of TTPVSS and three-tier satellite multicast security mechanism. Also, as a novel approach for cryptographic method, our mechanism uses multi-recipient signcryption scheme, which achieves confidentiality, authentication, unforgeability and non-repudiation together, more efficiently than classical sign-then-encrypt approaches. As a result, NAMEPS has many advantages for very large, dynamic and security critic satellite multicast systems.

6.1. Introduction to NAMEPS

NAMEPS is the latest satellite security mechanism among which we propose for SSMS in this thesis. NAMEPS also aims to provide same security and performance goals with TTPVSS and three-tier satellite multicast security mechanism. However, NAMEPS uses completely different approaches for structural design, key management and cryptographic technique aspects when compared to our previous security mechanisms. NAMEPS uses ELK key management protocol instead of LKH key management protocol. Also, NAMEPS uses signcryption based techniques, which provides advantages when compared to classical approaches and our previous security mechanisms.

NAMEPS uses N independent key distribution tiers to handle very large satellite multicast systems. NAMEPS especially addresses rekeying workload over satellite tiers, which is the major source of the performance problem in SSMSs. In NAMEPS, using N-tier structure, whenever a member join-leave event occurs, rekeying is realized on only related tiers and other parts of the system are not affected from modifications. This approach provides significant performance gain for satellite tiers as well as terrestrial tiers. Note that, most resource limited component of the SSMS is satellite tiers and NAMEPS especially reduces workload of the satellite tiers. Also, hierarchical structure of NAMEPS is compatible for military applications as well as commercial applications. NAMEPS uses ticketing and batch keying mechanisms, which also reduce workload of the satellite and terrestrial tiers. NAMEPS consists of GEO, MEO-LEO satellite tiers as satellite tier and terrestrial units (TU), major mobile units (MMU) and members tier as terrestrial tiers.

In SSMS, apart from structure, cryptographic primitives have critical importance for security and performance aspects. Cryptographic primitives must achieve major cryptographic goals such as confidentiality, authentication, integrity, unforgeability and non-repudiation. Moreover, in multi-tiered structures and especially for satellite networks, cryptographic workload may cause non-neglible delay and power consumption problems. In NAMEPS, to address these problems, as a novel approach, multi-recipient signcryption (SCS1M) methods are used for cryptographic key management. Signcryption is a relatively new concept, providing confidentiality, authentication, integrity, unforgeability and non-repudiation together more efficiently then classical sign-then-encrypt approaches (see chapter 3). Using SCS1M in SSMS provides significant advantages for both cryptographic workload and especially bandwidth consumption. As far as our concern, multi-recipient signcryption schemes have not been used in SSMS before.

6.2. Details of NAMEPS

6.2.1. Properties of NAMEPS

Our security mechanism uses N-tier structure to manage very large and dynamic satellite multicast groups. We use independency of tiers principle of the TTPVSS in our N-tier structure providing novel approaches for design and cryptographic method aspects. The major idea behind of NAMEPS is that, providing independency among tiers reduces rekeying workload of the satellite significantly. However, N-tier approach requires multiple encryptions between tiers which cause performance deteriorations. To address these problems, NAMEPS offers appropriate cryptographic methods that make N-tier approach feasible and secure.

Using independency of tiers principle, whenever a member join-leave event occurs, effects of the modifications are restricted on their related areas and remainder parts of the system are not affected from modifications. However, unlike TTPVSS, NAMEPS uses ELK protocol in each of its tier. ELK protocol has advantages for rekeying cost and size of packets when compared to LKH protocol (see chapter 2). Also, validation tickets are used which provide advantages for reliability. As mentioned in chapter 3, our security mechanism uses signcryption based cryptographic method SCS1M to transmit these keys securely. These improvements significantly reduce rekeying workload and number of keys which are stored on the satellite tier. Notice that, satellite tiers are the most resource limited components of the multicast system. Thus, it is critical to reduce the workload of these components. Responsibilities and properties of each tier are given below.

6.2.2. Detailed Description of NAMEPS

Before giving each steps for NAMEPS, we give notations that we use in NAMEPS. NAMEPS uses SCS1M as a major cryptographic method. Detailed notations and detailed descriptions for SCS1M are given in chapter 3.4. Other notations are given below:

$BBS(Blum - Blum - Shub)$: Cryptographically strong random number generator as in [53]. In our notations, BBS takes two types of parameters. First type of parameter determines upper or lower bound for number of elements that BSS generates. Second type of parameter is a seed value which triggers BBS for random number generation.

$SPNG$: Strong prime number generator,

Generator :Generates a generator g for related field,

n_s :Number of satellite in satellite tiers, l :Number of TU in SSMS, n_m :Number of MMU in SSMS, M : Bulk multicast data. Other notations are given when they are needed. Notice that, in NAMEPS, for bulk data transmission, symmetric key cryptography primitives are used. Block ciphers using appropriate modes such as OFB or CBC and stream ciphers can be used for these purposes [46].

6.2.2.1. Initialization, Key Distribution and Data Multicast in GEO Satellite Tier. In NAMEPS, top level of the hierarchy is GEO satellite. GEO satellite is responsible for generating all group keys and group key seeds for lower tiers. Also, GEO satellite actively involves bulk data multicast for regions that LEO and MEO satellites can not cover.

GEO satellite uses group key to distribute seeds and other group keys using SCS1M. GEO satellite determines group keys which LEO and MEO satellites use to securely communicate with TUs and MMUs. To do this, GEO satellite generates group key seeds st_i where $1 \leq i \leq n_s$. Each LEO and MEO satellite is assigned to a group key seed st_i. LEO and MEO satellites generate group keys $t_{i,j}$ using group key vector seed st_i where $j \geq l + n_m$. In here, $t_{i,j}$ denotes j'th group key that is used by i'th satellite in LEO and MEO satellite tiers. Major purpose of seeds values (vector st) is to provide batch keying. Using batch keying, instead of transmitting group keys one-by-one, only their seed values are transmitted, which provides significant

bandwidth advantages. Also, group key seeds for MMUs , sm_i *where* $1 \leq i \leq n_m$, are generated with same method and aim by GEO satellite. Note that, this approach is also effectively used in [13]. In NAMEPS, additionaly, validation tickets are used which provides important flexibility. Moreover, as a cryptographic method, SCS1M is used in each tier that provides additional advantages for batch keying, cryptographic workload and bandwidth consumption. However, in [13], cryptographic methods are different for each tier such as ECPVSS, IMC and ECMQV.

GEO satellite performs following steps:

1.1 Key generations: $p = SPNG(\geq 1024\ bits), g = Generator(p)$,

$(GK, x_a, y_a, st_i, sm_i, v_i) = BSS(=1, =1, =1, \geq n_s\ , \geq n_m, = n_s)$ and $y_i \leftarrow$(obtained from each LEO-MEO satellite) where $1 \leq i \leq n_s$. Here, v_i are random numbers, x_a is private and y_a is public key of GEO satellite.

1.2 GEO satellite signcrypts necessary keys and data for the lower tiers. Data packet M includes group key seeds st_i, sm_i and multicast data for GEO satellite m_{geo}.

$M = (st_i, sm_i, m_{geo})$, $h = H_{GK}(M)$, $c = E_{GK}(M, h)$,

$k_i = H(y_i^{v_i} \bmod p)$, $k_i \rightarrow (k_{i,1}, k_{i,2})$, $c_i = E_{k_{i,1}}(GK)$,

$r_i = H_{k_{i,2}}(M, h)$, $s_i = v_i(r_i + x_a)^{-1} \bmod q$.

1.3 GEO satellite multicast (c, c_i, r_i, s_i) where $1 \leq i \leq n_s$.

1.4 Whenever a satellite join-leave event occurs, key update is realized with ELK protocol using $k_{i,1}$ keys.

6.2.2.2. Key-Ticket Distribution and Data Multicast in LEO and MEO Satellite Tiers.

Each LEO and MEO satellite uses required (c, c_i, r_i, s_i) tuple to recover group key,

group key seeds and data. Private keys sx_i are used for public key generation. Public keys are sent to GEO satellite.

2.1 Key generations and transmission: $sx_i = BSS(= n_s)$, $y_i = g^{sx_i} \bmod p$, $GEO \leftarrow y_i$ where $1 \leq i \leq n_s$.

2.2 Unsigncryption Processes: Obtain tuple (c, c_i, r_i, s_i) and $k_i = H((y_a g^{r_i})^{s_i sx_i} \bmod p)$, $k_i \rightarrow (k_{i,1}, k_{i,2})$,

$GK = D_{k_{i,1}}(c_i)$, $w = D_{GK}(c)$ where $w = (M, h)$,

$if(((h == H_{GK}(M)) \wedge (r_i == H_{k_{i,2}}(w)))$ then signcryption is valid. Recover $M = (st_i, sm_i, m_{geo})$.

2.3 Each LEO and MEO satellite (i'th satellite in satellite tiers) uses group key seed st_i to generate group key vector $t_{i,j}$ as mentioned above. Whenever a TU join-leave event occurs, satellites use these group keys (elements of $t_{i,j}$) for group key update. For data multicast, satellites may directly pass m_{geo} or add their own multicast data such that $m_{geo} \subseteq m_{leo-meo}$. Also, LEO and MEO satellites generate validation tickets for usage of MMUs. Validation tickets are stored to response many-to-many multicast requests of MMUs. Note that, which TU is managed by which satellite is determined by an appropriate satellite internetworking mechanism like [56]. Each LEO and MEO satellite performs following steps:

$t_{i,j} = BBS(st_i)$, $1 \leq i \leq n_s$, $j \geq n_m$, $stc_i = BBS(\geq n_m)$. Public-private key generations are similar to step 1.1 in GEO satellite tier.

2.4 Each LEO and MEO satellite uses public keys of their related TUs (p', q', g', y_a') and performs following operations:

$GK' = t_{i,j}$, $M' = (stc_i, sm_i, m_{leo-meo})$,

$h' = H_{GK'}(M')$, $c' = E_{GK'}(M', h')$, $k_i' = H(y_i^{v_i'} \bmod p')$,

$k_i' \rightarrow (k_{i,1}', k_{i,2}')$, $c_i' = E_{k_{i,1}'}(GK')$,

$r_i' = H_{k_{i,2}'}(M', h')$, $s_i' = v_i'(r_i' + x_a')^{-1} \bmod q'$.

2.5 Each LEO and MEO satellite multicasts their (c', c_i', r_i', s_i') tuple where $1 \leq i \leq l$.

2.6 Whenever a TU join-leave event occurs, key update is realized with ELK protocol using $k_{i,1}'$ keys. LEO or MEO satellite realizes key update such that $t_{i,j} \rightarrow t_{i,j+1}$ and GEO satellite is informed for local group key update.

2.7 If many-to-many multicast requests come from MMUs or TUs, firstly LEO satellites, and if they are not available then MEO satellites validate tickets of MMUs or TUs. If ticket is valid then many-to-many multicast requests are performed. Validation mechanisms are mentioned at lower tiers. Note that, tickets are also used to assign one of the MMU if TU of that local region is not available.

6.2.2.3. Key-Ticket Distribution and Data Multicast in TU Tier. TUs are mainly responsible for decrypting required keys and multicast data coming from satellite tiers and multicast them in encrypted form using group key seeds sm_i for MMUs or members. Group key seeds and multicast data $m_{leo-meo}$ are recovered from (c', c_i', r_i', s_i') by each TU using SCS1M algorithm.

3.1 Public-private key pair generations are similar to the upper tiers. Each TU performs following steps:

$k_i' = H((y_a' g^{r_i'})^{s_i' s x_i'} \bmod p')$, $k_i' \rightarrow (k_{i,1}', k_{i,2}')$,

$GK = D_{k_{i,1}'}(c_i')$, $w' = D_{GK'}(c')$ where $w' = (M', h')$,

$if(((h' == H_{GK'}(M')) \wedge (r_i' == H_{k_{i,2}'}(w'))$ then signcryption is valid. Recover $M' = (stc_i, sm_i, m_{leo-meo})$.

3.2 Each TU generates ticket vector tc_i using ticket seed stc_i and generates $m_{i,j}$ group key vector using group key seed sm_i.

3.3 TU prepares data packets including $M'' = (m_{leo-meo}, tc_i)$ for its local MMU and member group and transmits these values to them. Like upper tiers, M'' is signcrypted using public keys of the MMUs and members that TU manages together with $m_{i,j}$.

3.4 Whenever a member or MMU join-leave event occurs, key update is realized with ELK protocol using $m_{i,j}$ group keys. TU realizes group key update such that $m_{i,j} \rightarrow m_{i,j+1}$ and LEO or MEO satellite is informed for group key update.

3.5 TUs evaluate many-to-many multicast requests of MMUs and members for local group without ticket requirement. For many-to-many multicast requests covering MMUs or members groups, which are related to other TU local groups, are evaluated using tc_i tickets. tc_i tickets are sent by MMUs using $m_{i,j}$ key of MMU. For this part of the SSMS, authentication may be provided by either $MAC(D_{m_{i,j}}(tc_i))$ using MAC (Message Authentication Code) or challenge response mechanisms which can utilize secret keys. Also, it is possible to use SCS1M for this purpose. Then, TU redirect validated multicast data and tickets to the LEO or MEO satellites. With same mechanism, LEO or MEO satellites decide whether many-to-many multicast requests are valid or not. If they are valid then, like original multicast data, many-to-many multicast data are done to related parts of the system.

6.2.2.4. Data Multicast and Member Management in MMU and Member Tiers. MMUs are used to provide reliability in terrestrial tiers. Especially for military applications, if TUs are not available, then using ticketing mechanism, one of the MMU is assigned as local group manager. Also, MMUs are required to support mobile light-weight mem-

bers that TU can not cover for various reasons. Many-to-many multicast requests of members are generally provided by MMUs using ticket mechanism via TUs, LEO or MEO satellite as mentioned at upper tiers. Details of Ad-Hoc network security mechanisms for MMU-Members tiers can be handled by various approaches such as [58] or [59].

6.3. Performance Analysis of NAMEPS

We analyze NAMEPS for two major criteria in details: Bandwidth consumption and computational workload. For these analyses, we firstly compare NAMEPS for cryptographic primitive aspect to other traditional approaches. Secondly, we analyze advantages of structural design and key management method used in NAMEPS focusing on two major criteria. Using these analyses, we give our simulation results comparing NAMEPS with pure implementation of LKH, OFT, ELK and TTPVSS.

Apart from these, NAMEPS provides advantages for storage requirement. N-tier structure of NAMEPS reduces number of keys that are stored in both satellite and terrestrial tiers. Each group manager in SSMS only stores keys for their related members. For satellites, they only store keys for their related TUs and MMUs ($\approx (2l + n_m)$). In pure implementations of LKH, OFT and ELK, satellite stores a unique key for each member, in total N keys. Thus, significant storage advantage is obtained. Also,when compared to TTPVSS, NAMEPS has advantage due to the satellite internetworking possibilities.

6.3.1. Advantages of SCS1M Usage in NAMEPS

In SSMS, traditional PKC (Public Key Cryptography) methods such as DH (Diffie-Hellmann), ECDH, factorization based and DLP based signature schemes together with their extensions of EC domains are used [36], [42]. NAMEPS uses SCS1M as major cryptographic primitive for PK data transmission. Thus, NAMEPS utilizes all advantages of signcryption based methods and especially SCS1M when compared to the traditional cryptographic primitives used in SSMS.

SCS1M has significant computational and communication overhead advantages when compared to classical DLP and factorization based signature approaches: For sender, SCS1M reduces number of the exponentiations and saves from computational cost by a factor larger than 50%. For recipient, SCS1M uses Shamir's fast exponentiations of the product of the exponentials and saves from computational cost by a factor 45%. Most significant gain is obtained for communication overhead. Note that, this criterion is the most important criterion for SSMS. For $|p| = 1024$ and $|q| = 160$, SCS1M provides communication overhead advantages up to 81%. Communication overhead gain increases when $|p|$ and $|q|$ increase. Saving for communication overhead is larger for RSA based key management approaches. Details can be found in chapter 3. Note that, DLP signcryption methods can be extended to EC domain [33]. Thus, advantages of NAMEPS are also valid for comparisons of ECC based traditional approaches.

In simulation results, we reflect aforementioned advantages of SCS1M to the classical approaches as cryptographic workload coefficient (CWC). CWC for SCS1M, TTPVSS and traditional methods are represented with c_1, c_2 and c_3 respectively. Following inequality holds: $c_1 \cong c_2 < c_3$.

6.3.2. Advantages of Structural Design and Properties of NAMEPS

In NAMEPS, N independent key distribution tiers provide major performance gain. In each tier, whenever a member join-leave event occurs, in order to provide forward and backward security, key update (rekeying) is realized on only related part of the tier and other parts of the SSMS are not affected from these processes.

Rekeying workload is the most important parameter that determines overall performance of SSMS. Rekeying workload is determined by number of join-leave event for certain time period, r, and cost of the rekeying operation. Cost of the rekeying operation depends on the cost of the applied key management protocol (ELK in each tier for NAMEPS) and cryptographic costs of the used cryptographic primitives (c_1, c_2 and c_3 for NAMEPS, TTPVSS and others, respectively). Using these facts, we calculate rekeying workload of the related group manager (satellite, TU or MMU)

according to the number of members it manages, number of rekeying, cryptographic cost, communication overhead and cost of the core key management protocol for key update operation. We refer total cost of the rekeying taking into consideration these parameters as TRBCC (Total Rekeying Bandwidth-Cryptographic Cost).

In NAMEPS, for each key update, ELK protocol is applied in related tiers. As mentioned in chapter 2, ELK provides smaller packet size and super-efficient member join when compared to many other protocols. For single member join, ELK does not cause rekeying workload. For member leave, ELK requires $log_k N \, |K\rceil$ where N is the number of members that ELK protocol is applied, k is the branching factor of the logical key tree, and $|K\rceil$ bit length of the ELK keys. However, in LKH and OFT protocols, both member join-leave operations causes rekeying workload larger than ELK leave event cost (join does not create cost) such that $(klog_k N - 1) \, |K|$ and $(log_k N - 1) \, |K|$, respectively. Note that, generally, $|K\rceil \leq |K|$.

Most resource limited component of the SSMS is satellite tiers. Thus, we specifically analyze workload of these tiers. Note that, analysis principles are same for terrestrial tiers. Each satellite manages a TU group having l_t TU on average. Whenever a TU join-leave event occurs, satellite realize rekeying with cost $c_1 \log_k l_t \, |K\rceil$. TU join events do not create workload. r_s is the number of rekeying for TU tier, $Pr(leave)$ is probability that occurred event is a leave event. Note that, TUs are generally static components, thus $r_s << r$ where r is number of join-leave events for all member (rekeying workload for overall system). This provides significant performance gain for NAMEPS when compared to pure implementations of LKH, OFT and ELK. Moreover, properties of SCS1M and abilities of the large satellite internetworking provide better batch keying, represented with m_1. This also reduces workload of the satellite tiers in NAMEPS when compared to TTPVSS mechanism, and pure implementations of LKH,OFT and ELK protocols. Moreover, in NAMEPS, a ticketing mechanism is used that make system more resistant against single point of failure problems. Also, with ticketing mechanism, MMUs can directly contact with satellites and can realize many-to-many multicast, even if their join-leave events does affect to the satellite tier. As a result, TRBCC for satellite tier is $c_1 \log_k l_t \, |K\rceil \, r_s \, \mathrm{Pr}(leave)/m_1$.

In TTPVSS, like NAMEPS, satellite is not affected from overall member join-leave rekeying workload, which is r. Thus, TTPVSS has a significant advantage to pure implementations of the LKH, OFT , and ELK protocols. TTPVSS uses LKH protocol in each of its tiers. When compared to NAMEPS, in TTPVSS, a satellite is responsible for all TUs . Also, due to no ticketing mechanism is used, MMUs are also managed by satellite which increases number of components that satellite is responsible for, $((l+n_m) >> l_t) \Leftrightarrow (r_{(l+n_m)} >> r_s)$. Moreover, for batch keying value of TTPVSS, $m_2 < m_1$. Thus, TRBCC for satellite in TTPVSS is $c_2 \log_k(l+n_m)\,|K|\,r_{(l+n_m)}/m_2$.

In pure implementations of LKH, OFT and ELK protocols, group manager is directly responsible for all members, thus parameter r affects the satellite. This causes massive workload over satellite. In addition to this, in previous security mechanisms, only TUs or MMUs involve core key management protocol cost. However, in this situation, $N >> n_m > l$ and significant workload occurs for satellite. In pure implementations, no specific batch keying or ticketing mechanism is used. Thus, TRBCC for satellite in LKH, OFT and ELK protocols are $c_3(k \log_k N - 1)\,|K|\,r$, $c_3(\log_k N + 1)\,|K|\,r$ and $c_3 \log_k N\,|K\rceil\,r \Pr(leave)$.

As a result, following TRBCC relation exists among pure implementations of LKH, OFT, ELK protocols and TTPVSS and NAMEPS mechanisms repectively:

$c_3(k \log_k N - 1)\,|K|\,r > c_3(\log_k N + 1)\,|K|\,r > c_3 \log_k N\,|K\rceil\,r \Pr(leave) >> c_2 \log_k(l+n_m)\,|K|\,r_{(l+n_m)}/m_2 >> c_1 \log_k l_t\,|K\rceil\,r_s \Pr(leave)/m_1$ where $r >> r_{(l+n_m)} >> r_s$ and $c_1 \cong c_2 > c_3$.

We can clearly see that, NAMEPS is the most efficient security mechanism among these security mechanisms for most important criteria.

6.3.3. Simulation Results

Simulation results are based on the evaluation of the satellite tiers for TRBCC measurements. In the first simulation (fig. 6.1), TRBCC responses of security mecha-

nisms for increasing members sizes for certain member rekeying value (r) are analyzed. We use rekeying ratio coefficient $rrc = 0.25$. In certain time period, $r = N{*}rrc$ rekeying occurs. TRBCC values of satellite tiers for $N = 10^5 \rightarrow 10^7$ are calculated. Rekeying behavior of r and TU-MMU join-leave events obey Poisson rule. Other parameters are taken according to the aforementioned criteria.

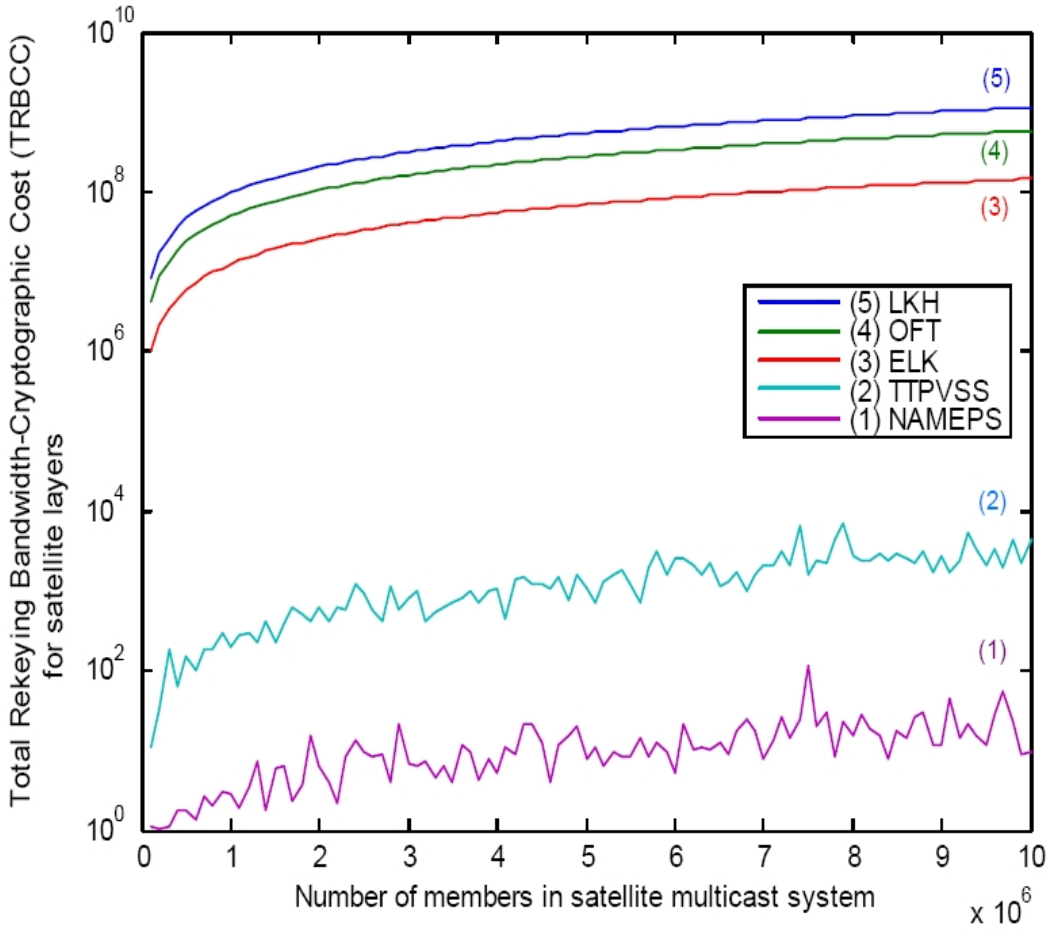


Figure 6.1. TRBCC comparison of NAMEPS for increasing member size

In second simulation (fig. 6.2), we analyze TRBCC responses of security mechanisms for increasing member dynamism, which is number of members join-leave for certain time period, for certain number of members. $N = 10^7$, $rrc = 10^{-3} \rightarrow 1$ and TRBCC workload of satellite tiers are observed. Randomized behaviors obey Poisson rule similar to above.

As a result, we can clearly see that NAMEPS has lowest workload among the pure implementations of LKH, OFT, ELK and TTPVSS . NAMEPS and TTPVSS

have significantly lower TRBCC workload than pure implementations of LKH,OFT and ELK. NAMEPS also has significant performance advantages to TTPVSS for aforementioned reasons. Simulation results show that NAMEPS can be applied extremely large multicast groups without having performance and security problem. Also, dynamism response of NAMEPS is very promising.

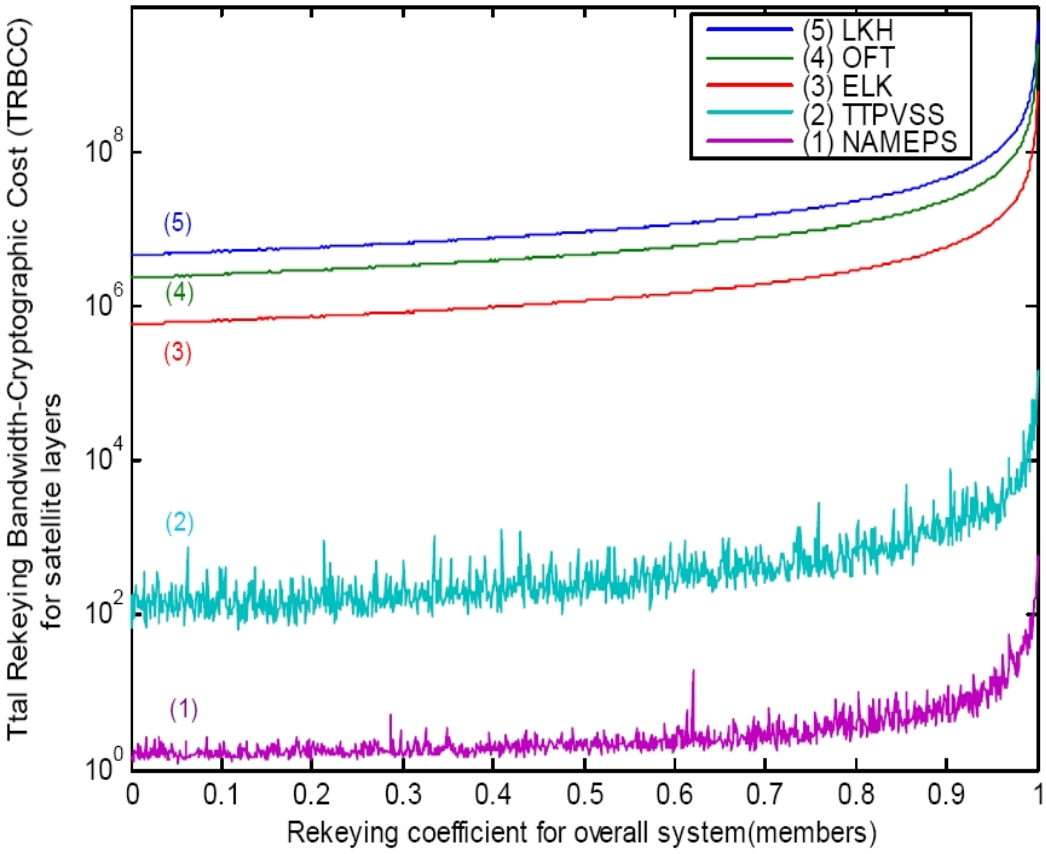


Figure 6.2. TRBCC comparison of NAMEPS for increasing member dynamism

6.4. Conclusions

In this study, we propose a new satellite multicast security mechanism using N-tiered structure based on multi-recipient signcryption schemes. NAMEPS uses N independent key distribution tiers to provide significant performance gain for especially satellite tiers. Effects of the group key update operations resulting from member join-leave events are restricted on only local parts of the related tiers. This approach significantly reduces rekeying workload and number of keys that are stored in satellite tiers as well as terrestrial tiers. Also, batch keying, validation ticket and N-tier satellite internetworking mechanisms are used that additionally reduces rekeying workload

and provides reliability for SSMS. NAMEPS uses ELK protocol in each of its tier providing advantages for rekeying costs and packet sizes. Apart from structural design, as a novelty, NAMEPS uses multi-recipient signcryption schemes (SCS1M) as a major cryptographic primitive for key management in SSMS. Using SCS1M, NAMEPS has all advantages of signcryption based schemes to the classical sign-then-encrypt approaches as well as utilizing additional benefits of multi-recipient version. We analyze and compare NAMEPS to the pure implementations of LKH, OFT and ELK and TTPVSS. Simulation results are given based on the analysis and comparison of these security mechanisms for TRBCC values. We can clearly see that NAMEPS is the most efficient security mechanism for major criteria such as bandwidth consumption, cryptographic workload and number of keys that are stored in components of the SSMS among mentioned security mechanisms. As a result, NAMEPS is especially suitable for very large and dynamic satellite multicast system requiring high security and provides significant advantages when compared to the some well-known security mechanisms.

7. HIMUTSIS: Hierarchical MUlti-Tier adaptive ad-hoc network security protocol (mechanism) based on SIgncryption type key exchance Schemes

Until this chapter, we use our structural design, hybrid key management approaches and cryptographic techniques in order to design efficient satellite multicast security mechanisms. However, as we have mentioned before, all of our approaches can be applied to any network system especially if the network has very large number of member and highly dynamic member join-leave characteristic. Firstly, our structural design and independency of tiers principle can be accepted as a generic design model and they can be applied to any hierarchical network. Also, our hybrid key management methods and cryptographic techniques can be used in both wired and wireless network types since they are designed to function especially for resource limited environments and they can be easily adapted to environments having better possibilities.

Taking into consideration these facts, we decided to use these principles for another challenging wireless network type to design an efficient network security mechanism. We have chosen Ad-hoc networks and specifically military Ad-hoc networks to design a new security mechanism. The reason is that, our approaches are especially suitable for military Ad-hoc networks since they are designed for highly resource limited, very large and dynamic and security critic networks.

Mobile Ad-hoc networks (MANETs), providing infrastructure-free wireless instant communication, play important role in tactical military networks. However, providing security in tactical military MANETs, having very large and dynamic structure without infrastructure support in hostile environments, is a very difficult task. In order to address security problems in tactical military MANETs, we propose a new HIerarchical MUlti-Tier adaptive ad-hoc network security mechanism based on SIgncryption type key exchange Schemes: HIMUTSIS. Our security mechanism makes contribution to the military MANETs in three major points: Structural design, cryptographic

methods used in military MANETs and key management techniques. Novel structural design of HIMUTSIS facilitates certification and key management procedures, provides flexibility and reduces cryptographic workload of the military MANETs. In HIMUTSIS, as a novelty, we offer to use DKEUTS (Direct Key Exchange Using TimeStamp) protocol providing security and performance advantages when compared to some traditional cryptographic methods. Also, multi-security level approach provides adaptive solutions for each tier of the HIMUTSIS. As a key management technique, HIMUTSIS uses hybrid key management approach which reduces rekeying workload of the networks significantly while minimizing single point of failure risk of the military MANET.

7.1. Introduction to HIMUTSIS

Mobile Ad-hoc networks (MANETs) are infrastructure-free wireless communication networks. MANETs are considered as ideal technology for instant communication networks in both military and civilian applications. Nowadays, tactical military networks are the main application area of MANETs. Tactical military networks, having critical operation environments, require very high security and performance together. Hostile environment of tactical military networks and infrastructureless-wireless characteristic of MANETs make these networks vulnerable to various attacks and compromises.

In this study, in order to answer these challenges, we propose a new HIerarchical MUlti-Tier adaptive ad-hoc network security mechanism based on SIgncryption type key exchange Schemes: HIMUTSIS. In HIMUTSIS, we make contributions to the military MANETs for three major points. These are design and security structure, cryptographic methods used in MANETs and key management techniques.

In HIMUTSIS, we use hierarchical multi-tier structure including novel approaches for design aspects. Two tiered UAV-MBN (Unmanned Aerial Vehicle-Mobile Backbone Networks) networks have been recently proposed for digital battlefields utilizing heterogeneous structure of military MANETs [60], [61]. In HIMUTSIS, as a novel approach, using same heterogeneity principle, we divide MBN tier into MBN1 and MNB2 tiers.

This approach significantly facilitates key management and certification procedures of military MANETs and reduces the threshold cryptography requirements. Particularly, when UAVs are not available in military MANETs, this structure provides flexibilities to the traditional approaches.

Many cryptographic methods have been proposed to secure MANETs [62]. In a secure MANET, availability, confidentiality, integrity, authentication, unforgeability and non-repudiation goals must be achieved [63]. In HIMUTSIS, as a novel approach, we use signcryption type key exhange scheme DKEUTS [30] as a major cryptographic method (see chapter 3). This method achieves all aforementioned cryptographic goals together while preventing network from some of the active attacks. Also, this method provides advantages for bandwidth and computational resource usage when compared to the classical methods. Apart from these, we propose a new multi-level security approach which provides high security for each tier while preventing system overloaded due to unnecessary cryptographic workloads.

In HIMUTSIS, we use hybrid key management techniques in order to scale very large and dynamic structure of military MANETs. We adapt independency of tiers principles of [16] and [13] to the MANETs. This approach significantly reduces workload of the rekeying which is required to provide forward and backward security. Also, single point of failure problem is minimized using hybrid key management structure.

The rest of the chapter is organized as follows. Section 7.2 gives related works and background for cryptographic methods in MANETs . Section 7.3 presents structural design of HIMUTSIS. Section 7.4 presents cryptographic techniques and security level approach of HIMUTSIS. Section 7.5 provides detailed steps of HIMUTSIS. Section 7.6 discusses analysis and performance comparison. The study is concluded in Section 7.7.

7.2. Related Works and Background

7.2.1. Cryptographic Techniques Used in Ad-hoc Networks

In order to achieve major cryptographic goals in Ad-hoc networks, many cryptographic methods utilizing public key cryptography and hybrid cryptography have been proposed. In Ad-hoc network, due to the lack of infrastructure, a static Trusted Third Party (TTP) may not be available. Thus, key exchange and key establishment schemes based on DH variants are frequently used for collaborative key exchange. Especially, for hierarchical key agreement in Ad-hoc networks, extending DH to the groups, Group Diffie-Hellmann GDH-1-2 [43] protocols are used. Moreover, Hybercube , Octopus and the Burmester-Desmedt protocols are used for hierarchical group key exchange [64]. In addition to these, key agreement protocols using generic password-based authenticated key exchange schemes and DH variants with extensions to the multi-party versions have been proposed in [62]. There are many other protocols using variants of these approaches [65].

Another important technique, which is frequently used in Ad-hoc network security, is the threshold cryptography. Threshold cryptography can be used to construct distributed public key management service to solve trusted certification problem. Using this approach, if some components of the system are compromised, single point of failure problem will not occur especially for certification issues. In [63], a distributed public-key management service for Ad-hoc networks has been proposed using these approaches.

7.3. Structural Design of HIMUTSIS

HIMUTSIS uses hierarchical multi-tier structure to secure and scale large and dynamic MANETs. Notice that, this structure is especially compatible with naturally existing hierarchical structure of the military networks. HIMUTSIS utilizes generic structure of hierarchical military MANETs such as [58] , [60], [61]. This structure consists of UAVs (Unmanned Aerial Vehicles), MBN (Mobile Backbone Networks)

and RGN (Regular Ground Nodes). Each UAV sets up and controls a MBN group having terrestrial mobile units in hierarchical manner. Also, each MBN sets up and controls RGN groups in hierarchical manner. In HIMUTSIS, we use a novel design approach and divide MBN into MBN1 and MBN2 tiers having different properties and duties. HIMUTSIS utilizes existing heterogonous structure and additional possibilities of MBN nodes in modern armies. This approach provides advantages for both security and performance aspects.

UAV-MBN1 tier consist of UAVs and MBN nodes having extensive communication capabilities such as long range missile batteries and mobile tactical centers. UAVs have a critical importance in modern battlefield communications. Small-low cost commercial off-the-Shelf (COST) radio equipment combined with powerful computer processing can be integrated on a UAV in order to form a multi-UAV, tactical-UAV and swarming UAV based MANET for both military and commercial applications. Notice that, as a reasonable assumption, both UAVs and MBN1 type nodes have advanced tamper resistant mechanism (for UAVs, an appropriate self-destruction mechanism can be applied) [61]. Thus, even if they are destroyed or captured by enemy, they will not comprise their cryptographic keys or certificates. Dividing MBN into MBN1 and MBN2 tiers, we extend advantages of tamper resistant mechanism into MBN tier and obtain some advantages for key management structure. In our mechanism, UAVs are mainly responsible for key distribution and certification processes as well as being bridge among MBN clusters for communication. Since number of MBN1 type nodes is limited, both storage and computational workload of UAVs are neglible.

MBN1-MBN2 is the second tier of our security mechanism. MBN2 nodes are generally mobile units used in classical UAV-MBN structure having high communication abilities. Special fighting units like trucks, tanks having beam-forming antennas can offer high-speed point-to-point direct wireless links in this tier [66]. MBN1-MBN2 tier utilizes possibilities of existing heterogeneous formation in MBN tier especially for armies having specialized ground units. This approach uses same heterogeneity principle which leads the creation of UAV-MBN networks. Notice that, our security mechanism can still function if MBN1 type nodes are not available. In this case, MBN2

type nodes will carry out duties of MBN1 type nodes using specific cryptographic techniques such as threshold cryptography in order to solve trust issues of certification [63].

MBN2-RGN is the third tier of our security mechanism. Each MBN2 controls RGNs including light weight equipped soldiers. In this tier, cryptographic algorithms are different from other tiers. Details are given in section 7.4 and 7.6.

7.4. Cryptographic Techniques and Security Level Structure of HIMUTSIS

In HIMUTSIS, we use a new multi-level security structure including cryptographic methods which have not been used in MANETs as far as our concern. In our mechanism, as a novel approach, we use signcryption based key exchange schemes as a major cryptographic method. In HIMUTSIS, we use DKEUTS based on SDSS1 type signcryption scheme [30]. Details of signcryption and DKEUTS protocol are given in chapter 3.

In HIMUTSIS, we suggest using a secure block cipher with appropriate modes such as AES in first and second tiers as symmetric encryption part of the signcryption process. Notice that, first tier of the structure particularly requires very high security. Thus, we suggest using at least 256 bit block cipher in this tier. Each signcryption scheme uses cryptographic hash functions to provide integrity and authentication. We suggest using at least 512 bit hash function such as SHA-512 [2]. Notice that, SHA-1 has been broken and threats for hash functions are in increase. Also, bit length of public key parameters should be hold as large as possible. We call security criteria determined for first tier as "Security level 1" (SL1). Same security approach, slightly reducing bit length of block ciphers, hash functions and public key parameters can be applied to the second tier. Notice that, security requirements are still high in second tier. We call this slightly reduced security level as "Security Level 2" (SL2).

In third tier, taking into consideration computational capabilities and communication scope of its nodes, we suggest using T-function [67] combined stream ciphers

such as ABC [68] as an alternative symmetric key cryptography method. Stream ciphers are especially preferred for their high speed encryption properties. Also, we use key transport protocol in this tier instead of key exchange protocol like DKEUTS or [13]. Bit length requirements are reduced and cryptographic methods are changed in this tier. We call this setting as "Security Level 3" (SL3).

Advantages of cryptographic techniques and security structure of HIMUTSIS including analysis are given in section 7.6.

7.5. Detailed Description of HIMUTSIS

In this section, we firstly present key management techniques used in HIMUTSIS. Then, we provide the details of HIMUTSIS.

7.5.1. Key Management of HIMUTSIS

Major principle behind key management techniques of HIMUTSIS is providing independency of tiers while preventing MANETs from performance deterioration and security problems. In order to provide forward and backward security (rekeying problem), we utilize independency of tiers and local rekeying principles for each tier of the HIMUTSIS. We use ELK protocol as a major key management protocol in each tier. As we have mentioned, ELK protocol has advantages for rekeying cost and size of the packets when compared to some well-known protocols such as LKH and OFT [9] (see chapter 2). Whenever a node join-leave event occurs in the theaters (active regions in military operations), ELK protocol is applied only related parts of the tier and other parts of the network are not affected from modifications. This provides significant performance gain and drastically reduces rekeying workload of the overall network. Notice that, some of these approaches have been effectively used in [13] and [70]. However, in HIMUTSIS, key management techniques are modified because structural design of [13] and [70] are completely different from HIMUTSIS and can not be applied directly. Apart from these, we also utilize batch keying mechanism of [16], [13] adapting them to the structure and requirements of HIMUTSIS.

We use certification procedures to provide authentication for public key's of the nodes in each tier. In [61], authors proposed an certification services for UAV-MBN networks. Our approach utilizes some principles of [61] but differs for cryptographic methods and key management techniques. A certificate is given to each theater by the manager of the theater in hierarchical manner. We represent certificates as $CERT_j^{l,i}$ including $(PK_j^{l,j}, T_{begin}, T_{end}, AddInf)$: Certificate of the $j'th$ unit in the $l'th$ tier and $i'th$ theater with denoted time intervals for public key $PK_j^{l,j}$. Notice that, inter-theater migration of nodes can be succeeded by DKEUTS key exchange mechanism utilizing the approach used in [58]. Due to space limitations, we will not give details of the certification and inter-theater migration procedures.

7.5.2. Detailed Description of HIMUTSIS

We adapt DKEUTS scheme to our multi-tier hierarchical military MANET structure. Following notations are used:

$K_{i,j}^{s,d}$:Directed secret key in key exhange procedure. It is transmitted from $i'th$ source s_i to $j'th$ destination d_j.

Source or destination can be following node types, $u : UAV$, $m_1 : MBN1\ node$, $m_2 : MBN2\ node$. All other internal keys adapted from DKEUTS obey same notation rules.

$KT_i^{\gamma_l}$: This is intra-theater group communication key generated by theater manager. γ_l represents theater level and index i denotes index of the group manager in level l.

$su_{i,j}$: Seed value transmitted from $i'th$ theater manager to $j'th$ node in that theater. These seed values are used for moderate-time batch keying purposes.

SKG (Symmetric Key Generator): Generate keys obeying security level which is sent as a parameter to the function. Also, it may take a seed value to generate keys

with related security level.

SGNKG (*Si*gncryption Key Generator): Similar to *SKG* but generates signcryption related paramteres such as,

p :Large prime number, q :A large prime factor for $(p-1)$, g :Generator of the group with order q modulo p and other signcryption parameters: $xa_{i,j}^{s,d}, xb_{i,j}^{s,d}$ are private parameters and $ya_{i,j}^{s,d}, yb_{i,h}^{s,d}$ are public parameters of signcryption based schemes.

H :Unkeyed cryptographic hash function, $H_{K_{i,j}^{s,d}}$:Keyed cryptographic hash function, $(E-D)_{K_{i,j}^{s,d}}$:Symmetric encryption-decryption function.

n, n_{type}, n_{type_i} : Number of total nodes, number of *type* nodes and number of *type* nodes in the $i'th$ theater in MANET, respectively. M : Messages. Other notations are given when they are needed.

7.5.2.1. UAV-MBN1 Tier:. **7.5.2.1.1 Key Generation:**

UAVs: $(su_{i,j}, KT_i^{\gamma_1}, K_{i,j}^{u,m_1}, x_{i,j}^{u,m_1}) = SKG(SL1)$, obtain $y_j^{m_1,u}$ from MBN1 nodes and $(p_i, q_i, g_i, xa_{i,j}^{u,m_1}) = SGNKG(SL1)$.

MBN1 Nodes: $(K_{j,i}^{m_1,u}, x_{j,i}^{m_1,u}) = SKG(SL1)$, obtain y_i^{u,m_1} from UAVs and $xb_{j,i}^{m_1,u} = SGNKG(SL1)$ where $1 \leq i \leq n_u$, $1 \leq j \leq n_{m_{1_i}}$ for each i and $l = 1, 2$.

7.5.2.1.2 DKEUTS Steps:

UAVs Key Transport: $(k_{1,i,j}^{u,m_1}, k_{2,i,j}^{u,m_1}) = H((y_i^{m_1,u})^{x_{i,j}^{u,m_1}} \bmod p_i)$ and each UAV gets their current time-stamps $TS_{i,j}^{u,m_1}$.

$c_{i,j}^{u,m_1} = E_{k_{1,i,j}^{u,m_1}}(K_{i,j}^{u,m_1}, TS_{i,j}^{u,m_1})$, $r_{i,j}^{u,m_1} = H_{k_{2,i,j}^{u,m_1}}(K_{i,j}^{u,m_1}, TS_{i,j}^{u,m_1}, CERT_j^{\gamma_{l,i}})$,

$s_{i,j}^{u,m_1} = x_{i,j}^{u,m_1}(r_{i,j}^{u,m_1} + xa_{i,j}^{u,m_1})^{-1} \bmod q_i$ and UAVs transmit $(c_{i,j}^{u,m_1}, r_{i,j}^{u,m_1}, s_{i,j}^{u,m_1})$

tuples to the MBN1 nodes.

MBN1 Nodes Verification: $(k_{1,i,j}^{u,m_1}, k_{2,i,j}^{u,m_1}) = H((y_{i,j}^{u,m_1} \cdot g_i^{r_{i,j}^{u,m_1}})^{s_{i,j}^{u,m_1} \cdot xb_{j,i}^{m_1,u}} \bmod p_i)$, $(K_{i,j}^{u,m_1}, TS_{i,j}^{u,m_1}) = D_{k_{1,i,j}^{u,m1}}(c_{i,j}^{u,m_1})$ then perform following control:

$If(Freshness(TS_{i,j}^{u,m_1} == true) \wedge (H_{k_{2,i,j}^{u,m_1}}(K_{i,j}^{u,m_1}, TS_{i,j}^{u,m_1}) == r_{i,j}^{u,m_1}))$ then accept else reject.

MBN1 Nodes Key Transport: $(k_{1,j,i}^{m_1,u}, k_{2,j,i}^{m_1,u}) = H((y_i^{u,m_1})^{x_{j,i}^{m_1,u}} \bmod p_i)$ and each MBN1 node gets their current time-stamps $TS_{j,i}^{m_1,u}$.

$c_{j,i}^{m_1,u} = E_{k_{1,j,i}^{m_1,u}}(K_{j,i}^{m_1,u}, TS_{j,i}^{m_1,u})$, $r_{j,i}^{m_1,u} = H_{k_{2,j,s}^{m_1,u}}(K_{j,i}^{m_1,u}, TS_{j,i}^{m_1,u}, CERT_j^{\gamma_{l,i}})$,

$s_{j,i}^{m_1,u} = x_{j,i}^{m_1,u}(r_{j,i}^{m_1,u} + xa_{j,i}^{m_1,u})^{-1} \bmod q_i$ and UAVs transmit $(c_{j,i}^{m_1,u}, r_{j,i}^{m_1,u}, s_{j,i}^{m_1,u})$ tuples to the MBN1 nodes.

UAVs Key Verification: $(k_{1,j,i}^{m_1,u}, k_{2,j,i}^{m_1,u}) = H((y_{j,i}^{m_1,u} \cdot g_i^{r_{j,i}^{m_1,u}})^{s_{j,i}^{m_1,u} \cdot xa_{i,j}^{u,m_1}} \bmod p_i)$,

$(K_{j,i}^{m_1,u}, TS_{j,i}^{m_1,u}) = D_{k_{1,j,i}^{m_1,u}}(c_{j,i}^{m_1,u})$ then perform following control:

$If(Freshness(TS_{j,i}^{m_1,u} == true) \wedge (H_{k_{2,j,i}^{m_1,u}}(K_{i,j}^{u,m_1}, K_{j,i}^{m_1,u} TS_{j,i}^{m_1,u}) == r_{j,i}^{m_1,u}))$ then accept else reject.

7.5.2.1.3 Complete Key Exchange:

Both UAVs and MBN1 nodes: $K_{i,j}^* = K_{i,j}^{u,m_1} \oplus K_{j,i}^{m_1,u}$ then unique shared key pairs $K_{i,j}^*$ have been created among UAVs and MBN1 nodes. As an optional step:

UAV: $tag_{i,j}^{u,m_1} = MAC_{K_{i,j}^*}(TS_{i,j}^{u,m_1})$ and send tags to the MBN1 nodes. MBN1 nodes verify tags $if(MAC_{K_{i,j}^*}(TS_{i,j}^{u,m_1}) == true)$.

7.5.2.1.4 Secure Communication and Key Transmission:

UAVs: $M_{i,j}^{u,m_1} = (KT_i^{\gamma}, su_{i,j})$, $M_{i,j}^{*} = E_{K_{i,j}^{*}}(M_{i,j}^{u,m_1})$, $M_i^{'} = E_{KT_i^{\gamma}}(m_i^{\gamma})$ where $M_{i,j}^{u,m_1}$ message includes intra-theater communication keys and batch keying seeds for each nodes. For each nodes, $M_{i,j}^{u,m_1}$ are encrypted with shared keys $K_{i,j}^{*}$.

MBN1: $M_{i,j}^{u,m_1} = D_{K_{i,j}^{*}}(M_{i,j}^{*})$ and recover $KT_i^{\gamma}, su_{i,j}$ keys from $M_{i,j}^{*}$. Now, each MBN1 nodes in related theaters have intra-theater communication keys KT_i^{γ} . Using these, , $m_i^{\gamma} = D_{KT_i^{\gamma}}(M_i')$ and each MBN1 nodes obtain intra-theater message m_i^{γ}. MBN1 nodes can communicate with their UAV using $K_{i,j}^{*}$.

7.5.2.1.5 Member-Join Leave:

Whenever a MBN1 node join-leave event occurs in a UAV theater, UAV applies ELK key update rules using $K_{i,j}^{*}$ unique keys of each MBN1 nodes.

7.5.2.2. MBN1-MBN2 Tier:. In this tier, like upper tier, DKEUTS key exchange is realized among MBN1 and MNB2 nodes. MBN1 may use same batch keying mechanisms. Key generation and parameter bit lengths obey SL2 criteria. As an optional step, MBN1 nodes can generate their directed unique keys $K_{i,j}^{m_1,m_2}$ using $su_{i,j}$ seeds. Then, each key update in MBN1-MBN2 tier can be tracked by UAVs. If this is not desired, key generation rules for these keys can be done similar to the upper tier. Due to space limitation, we give summarized version of steps of this tier.

Following notations are used: $CRS_{i,j}^{m_1,m_2}$ denotes $(c_{i,j}^{m_1,m_2}, r_{i,j}^{m_1,m_2}, s_{i,j}^{m_1,m_2})$ and $CRS_{j,i}^{m_2,m_1}$ denotes $(c_{j,i}^{m_2,m_1}, r_{j,i}^{m_2,m_1}, s_{j,i}^{m_2,m_1})$ tuples. DKEUTS parameter transport and verification procedures are represented with $DKEUTST$ and $DKEUTSV$.

Key Generation: $K_{i,j}^{m_1,m_2} = SKG(SL2, su_{i,j})$ and with this UAVs $su_{i,j}$ seeds are used for key generation. Then, $(KT_{l,i}^{\gamma}, x_{i,j}^{m_1,m_2}, x_{j,i}^{m_2,m_1}, K_{j,i}^{m_2,m_1}) = SKG(SL2)$,

$(p_i^{*}, q_i^{*}, g_i^{*}, xa_{i,j}^{m_1,m_2}, xb_{j,i}^{m_2,m_1}) = SGNKG(SL2)$ and $t^{*} = (p_i^{*}, q_i^{*}, g_i^{*})$ where $1 \leq i \leq n_{m_1}$, $1 \leq j \leq n_{m_{2_i}}$ for each i and $l = 2, 3$.

Adapted DKEUTS Steps:

$$CRS_{i,j}^{m_1,m_2} = DKEUTSG^{m_1,m_2}(y_{j,i}^{m_2,m_1}, TS_{i,j}^{m_1,m_2}, xa_{i,j}^{m_1,m_2}, x_{i,j}^{m_1,m_2}, t^*),$$

$$(TS_{i,j}^{m_1,m_2}, K_{i,j}^{m_1,m_2}) = DKEUTSV^{m_2,m_1}(y_{i,j}^{m_1,m_2}, xb_{j,i}^{m_2,m_1}, CRS_{i,j}^{m_1,m_2}),$$

$$CRS_{j,i}^{m_2,m_1} = DKEUTSG^{m_2,m_1}(y_{i,j}^{m_1,m_2}, TS_{j,i}^{m_2,m_1}, xb_{j,i}^{m_2,m_1}, x_{j,i}^{m_2,m_1}, K_{i,j}^{m_1,m_2}, t^*),$$

$$(TS_{j,i}^{m_2,m_1}, K_{j,i}^{m_2,m_1}) = DKEUTSV^{m_1,m_2}(y_{j,i}^{m_2,m_1}, xa_{i,j}^{m_1,m_2}, CRS_{j,i}^{m_2,m_1}),$$

$K'_{i,j} = K_{i,j}^{m_1,m_2} \oplus K_{j,i}^{m_2,m_1}$ where $K'_{i,j}$ unique shared key pairs, which have been created among MBN1 and MBN2 nodes. Similar to upper tier, $KT_i^{\gamma_2}$ intra-theater group communication key has been transmitted using $K'_{i,j}$. Then, intra-theater message $m_i^{\gamma_2}$ can be securely distributed using $KT_i^{\gamma_2}$.

Member Join-Leave: Whenever a MBN2 node join leave event occurs in a MBN1 theater, MBN1 applies ELK key update rule. There are two options for key update. If batch keying mechanism is used, theater manager (MBN1) nodes generate its directed secret key using $su_{i,j}$. Then, whenever a key update occurs, instead of obtaining new key seeds from UAVs, MBN1 nodes uses $su_{i,j}$ seeds to generate next seed value and inform this process to the UAV. Details about this mechanism can be found in [13] and [70].

7.5.2.3. MBN2-Regular Ground Node Tier:. In this tier, we suggest using SL3 criteria. As discussed in section 7.4, instead of joint key exchange, a key transport mechanism like [16] or multi-recipient signcryption scheme like [70] can be used. Benefits of this approach are given in section 7.6.

7.6. Performance Analysis of HIMUTSIS

7.6.1. Properties of Cryptographic Methods Used in HIMUTSIS:

Major cryptographic method used in our security mechanism is DKEUTS key exchange protocol. DKEUTS protocol is based on signcryption and it inherently utilizes all security properties of signcryption schemes. We summarize benefits of DKEUTS protocol to the some traditional cryptographic methods below:

- DKEUTS protocol achieves confidentiality, authentication, integrity, unforgeability and non-repudiation. Notice that, many traditional cryptographic methods can not achieve these five major cryptographic goals together. Freshness of the messages is provided by either time-stamps or nonces. This provides security against some active attacks.
- Signcryption, when compared to the classical sign-then-encrypt approach, has both computational and bandwidth advantages. When compared to the sign-then encrypt approach using Shcnorr and and El-Gamal signature, in average, signcryption provides 58% computational and 78 % communication overhead advantages for RSA based signatures [30].
- We denote cryptographic advantages of the DKEUTS protocol for both bandwitdh and computational effort as c_{sgn} and cryptographic cost of traditional methods as c_{trd} .

Apart from benefits coming from DKEUTS, HIMUTSIS has a multi-security level structure which provides many advantages when compared to the traditional approaches. In traditional approaches, generally, all components of the network are enforced to use same cryptographic methods without regarding their heterogeneous computational and storage possibilities. In HIMUTSIS, we use three different security levels having two different cryptographic approaches. In first and second tiers, joint key exchange method DKEUTS has been preferred instead of key transport protocol used in [70] or [16] . The reason is that, especially in the first tier, trust level (military ranks and rights, possibilities and hardness of the capturing of the nodes can be criteria) and

computational availability of UAV and MBN1 nodes are close to each other both of them having tamper resistant possibilities. Thus, both parties of the communication should have right to determine their unique key pairs $K^*_{i,j}$ in equal manner. Similarly, it is also reasonable for MBN1-MBN2 nodes to realize joint key exchange having equal rights. Security level of each tier depends on three major criteria: Communication scope, importance of the information and computational-storage possibilities of the nodes in that tier. In first tier, communication scope is significantly larger than other tiers. Importance of the information context in this tier is very high since it is the top level of the hierarchy and nodes of this tier are critical components of the armies (UAVs, tactical centre etc). Also, computational possibilities of UAV-MBN1 nodes are better than other tiers. Thus, highest security parameters having very large bit sizes are used. In second tier, significance criteria are slightly lower than first tier and bit length of security parameters is reduced. In third tier, RGNs, which have low communication possibilities, consist of majority of the tier. Moreover, communication density is expected to be high while scope of the communication is significantly smaller than other tiers. Thus, cryptographic algorithms focusing on high speed and low storage requirements such as stream ciphers can be more appropriate for this tier. Also, since there is important possibilities and trust difference among MBN2 and RGNs, we use key transport protocol similar to [70] or [16]. Apart from these, in third tier, using a different approach, we suggest using T-function supported stream ciphers [69].

7.6.2. Structural Design and Key Management Properties of HIMUTSIS

Structural design of HIMUTSIS provides advantages for security, scalability and performance aspects. We give properties of structural design and key management techniques of HIMUTSIS below:

- HIMUTSIS utilizes heterogenic structure of MBN tier and divides MBN tier into MBN1 and MBN2 tiers. MBN1, having tamper resistant properties, facilitates certification procedures when central manager of the theater is destroyed. Duplication of the certificates of the UAVs is now possible for MBN1 tier and this approach reduces threshold cryptography requirement. Notice that, if needed,

threshold mechanism can still be applied to MBN2 tier.

- Main principles behind of the hybrid key management techniques of the HIMUTSIS are below:
  - Pure decentralized structures are not suitable for naturally hierarchical and central entity based military applications.
  - Pure centralized structures cause SPoF problems. This problem becomes much severe for highly dynamic military MANETs where survability of nodes can not be guaranteed.
  - HIMUTSIS divides very large and dynamic MANET into subgroups like decentralized approaches in order to prevent SPoF . At the same time, HIMUTSIS uses centralized key management technique in each theater in order to provide scalability and forward-backward security. Similar approach is also used in [58].

In HIMUTSIS, significant performance gain is obtained from independent multi-ELK-theater approach. In each theater, whenever a node join-leave event occurs, in order to provide forward and backward security, key update (rekeying) is realized on only related part of the theater using ELK and other parts of the system are not affected from these processes. This approach minimizes rekeying workload of MANET and provides significant performance gain. We define ORW (Overall Rekeying Workload) measurement for cost of the rekeying operation. Measurement is defined according to the three major criteria: Number of join-leave event for certain time period in certain scope of the network, r_{scope}, cost of the rekeying mechanism used in network, $c_{protocol}$ (also related with number of members affected from rekeying), and cost of the cryptographic methods used in key management, c_c. ORW can be determined approximately as $r_{scope} \cdot c_{protol} \cdot c_c$.

We compare HIMUTSIS to pure centralized approaches using LKH, OFT and ELK in the context of their ORW measurements. In pure centralized approach, rekeying of all network components is done by only central entity. Thus, for aforementioned protocols, number of affected nodes is represented by n, which is all nodes in the network. In HIMUTSIS, for each node join-leave, only related theater is affected. Thus,

number of affected nodes is represented with thr where $thr << n$. Also, number of rekeying in a single theater, r_{thr}, is much smaller than rekeying of all network, r , for certain time period and $r_{thr} << r$. m denotes benefits coming from batch keying and this factor additionally reduces ORW of the HIMUTSIS. k denotes branching factor of the logical key tree. Detailed cost analysis of LKH, OFT and ELK protocols can be found in [5], [8]. Comparison results are given at Table 7.1 below.

Table 7.1. Comparison of HIMUTSIS with some Pure Centralized Key Management Protocol for ORW

	ORW	Storage Cost	SPoF Problem
LKH	$c_{trd}O(k\log_k n - 1)r$	$\mathrm{O}(\log_k n \,\lvert K\rvert)$	Yes
OFT	$c_{trd}O(\log_k n)r$	$\mathrm{O}(\log_k n \,\lvert K\rvert)$	Yes
ELK	$c_{trd}O(\log_k n)\Pr(leave)r$	$\mathrm{O}(\log_k n \,\lvert K\rvert)$	Yes
HIMUTSIS	$c_{sgn}O(\log_k thr)\Pr(leave)r_{thr}m^{-1}$	$\mathrm{O}(\log_k(thr) \,\lvert K\rvert)$	No
PDC	Trust problems, not suitable for military applications		No

As we have seen, HIMUTSIS has significant advantages over the pure implementation of the centralized approaches. These advantages stem from the decentralized properties of HIMUTSIS and both $r_{thr} << r$ (most important gain) and $thr << n$. Thus, performance of HIMUTSIS is better than pure implementation of these mechanisms. Also, in pure centralized approach, SPoF problem occurs while this problem is minimized in HIMUTSIS. When compared to Pure Decentralized Approach (PDA), HIMUTSIS is more appropriate for military MANETs as discussed above.

7.7. Conclusions

In this study, we have proposed a new hierarchical multi-tiered adaptive Ad-hoc network security mechanism based on signcryption type key establishment schemes: HIMUTSIS. HIMUTSIS brings novelties for structural design of military MANETs, cryptographic methods used in MANETs and usage of hybrid key management approaches. Structural design of HIMUTSIS consists of UAV, MBN1-MBN2 and RGN tiers in hierarchical manner. Structural design of HIMUTSIS differs from traditional UAV-MBN networks with MBN1-MBN2 tiers utilizing heterogeneity of MBN tier and

tamper resistant possibilities of MBN1 nodes. This approach makes possible to give centralized certification rights of the UAVs' to the MBN1 (tamper resistant) which reduces threshold cryptography requirement and facilitates certification procedures. HIMUTSIS uses multi-security level approach for its tiers applying high security parameters with DKEUTS signcryption type key exchange in its two tiers and stream ciphers based key transport schemes in the third tier. Adapted DKEUTS provides all security and computational-bandwidth advantages of signcryption schemes to the HIMUTSIS when compared to the traditional cryptography approaches. Also adapted DKEUTS prevents MANETs from some active attacks. HIMUTSIS utilizes hybrid key management techniques to the military MANETs. HIMUTSIS divides military MANETs into hierarchical tiers and theaters using decentralized approach preventing system SPoF problem. In each theater, HIMUTSIS uses ELK centralized protocol to scale large and dynamic military MANETs. As a result, HIMUTSIS is especially suitable for very large and dynamic military MANETs requiring very high security and performance.

8. IMC (Improved Merkle Cryptosystem)

In this chapter, we focus on a completely different topic about security systems. Up to now, we have focused on structural design of wireless networks, key management techniques and novel cryptographic approaches to design efficient security mechanisms. However, in this study, we focus on creating a cryptosystem which significantly improves an existing cryptosystem by means of both security and performance aspects. Notice that, with these improvements, it is also possible to use this cryptosystem to design key exchange protocols and utilize it for designing secure protocols.

Merkle Cryptosystem (MC) is the first cryptosystem which introduces general concept of the public key cryptography. In this study, we propose Improved Merkle Cryptosystem (IMC), which has significant security advantages over both MC and a variant of MC (VMC). In IMC, cryptographic hash functions and a new puzzle structure are used together in order to increase the security of MC and VMC. The key agreement value, which is sent as clear text in VMC, is hidden using cryptographic hash function in IMC. Also, in order to increase security of the key agreement value, auxiliary keys are used. Notice that, in IMC, computational advantages of VMC remain unchanged while its security is increased. Utilizing computational advantages of VMC, IMC has also security and storage advantages over original MC. It is shown that, with these improvements, IMC can provide as high security as some well-known public key cryptosystems while MC and VMC can not provide same security due to the performance problems.

8.1. Introduction to IMC

Public key cryptography made significant impact on secure and authenticated communication systems [36]. As we have mentioned in chapter 3, many different public key cryptography algorithms have been developed based on different mathematical approaches [71]. RSA, which is based on factorization of large numbers into prime factors and El-Gamal cryptosystem, which is based on hardness of DLP, are well-known

and fundamental public key cryptosystems [42]. Also, ECC [37] which is based on DLP over EC is one the most widely used cryptosystem utilizing DLP. Apart from these, new public key cryptosystems such as NTRU (N-th degree TRUncated polynomial ring) [72], which is based on lattice problem, have also been proposed.

However, the first cryptosystem, which provides a solution to the secure communication problem over insecure channels without pre-established secrets, is Merkle Cryptosystem (MC) [54]. In MC, communicating parties use 'puzzles', which are feasible for them to solve but infeasible for an attacker to solve. A Variant of MC (VMC) [55] utilizes MC with block ciphers and uses a different puzzle generation technique from MC. VMC method has some advantages over original MC.

In this study, we propose Improved Merkle Cryptosystem (IMC) that increases the security of the both MC and VMC. In VMC, the index, which is used for key agreement, is sent in clear text. This approach causes significant security degradation. In IMC, we use a different puzzle structure and cryptographic hash functions to increase security of VMC. IMC utilizes puzzle generation method of VMC but uses auxiliary key to increase security of messages transmitted over network. Also, in order to hide key agreement value, we use cryptographic hash functions. Thus, adversary can not understand which puzzle communicating parties agree on (auxiliary keys increases security of the hashed key agreement value). In addition to this, computational advantages of the VMC remain unchanged while its security is significantly increased. Shifting computational advantages of communicating parties to the overall system security, IMC can also provide higher security than original MC. Moreover, puzzle structure of IMC provides storage advantages over original MC method. We also show that, with these improvements, IMC can provide as high security as some well-known public key cryptosystems while MC and VMC can not provide same security due to performance issues.

The rest of the chapter is organized as follows: In Section 8.2, we discussed MC and VMC algorithms together with their security analysis. In Section 8.3, we present our IMC algorithm and its properties. In Section 8.4, we give detailed analysis of

IMC algorithm and compare IMC to MC and VMC. Also, comparison of IMC for various criteria to some well-known public key cryptosystems and MC-VMC is given. In Section 8.5, we present conclusions and future works.

8.2. MC and VMC

8.2.1. Merkle Cryptosystem (MC)

Merkle Cryptosystem (MC), also known as Merkle Puzzle, is the first cryptosystem having public key cryptography properties [54]. Suppose that, Alice and Bob want to secretly communicate over an insecure channel without pre-established secrets. Alice creates a set of puzzles that are feasible to solve for Bob. These puzzles are derived from secre$t$ values using secret keys that are short enough such that Bob can realize brute force attack on them. Each puzzle contains a session key that will be used for future communication and a pseudo-index which makes possible secret key establishment. In addition, each puzzle is added the required redundancy that allows Bob to perform the brute force attack. Bob selects one of the puzzles and performs a brute force attack on it. Bob stops brute force attack when he detects recognizable redundancy value. Bob recovers pseudo-index and session key from solved puzzle and sends pseudo-index to Alice. Alice searches this pseudo-index in her pseudo-index list and find corresponding real index which Bob has chosen. Consequently, Bob and Alice agree on a secret session key which corresponds to the selected real index. Adversary (Oscar) only observes pseudo-index, which does not reveal any information about which key Bob has chosen. Thus, adversary has to make brute force attack to all puzzles. Here, Bob makes brute force attack only one puzzle while adversary makes brute force attack to all puzzles.

Apart from being the first cryptosystem which introduces general concept of the public key cryptography, principles of MC are used in many security applications. For instance, puzzle principle of MC is used in time-lock puzzles [73]. In time-lock puzzles, the idea is that a secret is transformed in such a way that any machines, running continuously, take at least a certain amount of time to solve the puzzles in order to recover the secret. This minimum amount of time is the relative release time with

respect to the start of solving the puzzle and could be different for different machines [74]. In addition to this, puzzle concept of MC is used to combat against junk mails [75] and is used to prevent DoS (Denial of Service) attacks utilizing client puzzles [76].

Notations, which are used in MC, are given below:

P : Public key vector (puzzles),

K : Secret key vector that is used to generate P,

K_s : Session key vector. $P_i \in P$, $K_i \in K$ and $K_{s_i} \in K_s$ for $1 \leq i \leq \overline{N}$ where $\overline{N} = 2^m$.

m: The parameter which, determines number of elements in the P, K and K_s vectors.

In MC, bit length of the secret key is represented by $n' = |K_i|$ and bit length of the single puzzle is represented by $t = |P_i|$. $(E - D)_K$: Symmetric encryption and decryption functions using secret key K.

$\overline{r}_{ind}$: Index number used for key agreement.

S : The recognizable redundancy value.

Original version of MC is described below:

1. Alice generates puzzles $P_i = E_{K_i}(S||\overline{r}_i||K_{s_i})$ for $1 \leq i \leq \overline{N}$ where $\overline{N} = 2^m$. Alice sends vector P to Bob.

2. Bob selects one of the puzzle say j'th puzzle and realizes a brute force attack to P_j. When brute force attack is completed, Bob decrypts the puzzle $(S||\overline{r}_j||K_{s_j}) = D_{K_j}(P_j)$. Bob verifies S and recover $\overline{r}_j$ and K_{s_j}.

3. Bob sends index $\overline{r}_j$ to Alice in clear text. Notice that this index $\overline{r}_j$ is a pseudo-index and only Alice knows which real index corresponds to pseudo-index $\overline{r}_j$. Suppose that pseudo-index $\overline{r}_j$ corresponds $i'th$ puzzle. Then, Alice knows that Bob has chosen $i'th$ puzzle P_i from P.

4. Alice and Bob agree on the secret session key K_{s_i} and use this key for future communication.

In this system, symmetric encryption function can be an appropriate block cipher such as DES or AES [45]. Notice that, n' i.e., that is the bit length of the K_i should be selected carefully. It should allow Bob to realize a brute force attack on P_i but should not be so small such that it weakens the whole cryptosystem. In the first version of the MC, n' is selected as 20 bits. Also, some versions select $n' = m$ so that number of puzzles and bit length of the single puzzle are equal to each other. In MC, Oscar can listen the communication channel and observe index $\overline{r}_j$. However, since index $\overline{r}_j$ does not reveal which puzzle Bob chooses, Oscar has to realize brute force attack to whole puzzles in order to understand which puzzle Bob has chosen. In original MC, bit length of single puzzle P_i is $|t| = (S||r_i||K_{s_i})$ where $t > n'$. This property increases storage requirements of the original MC. Notice that, reasonable bit length of $n' \leq 50$. Complexity of the MC is summurized at Table 8.1.

Table 8.1. Computational and Storage Complexity of MC

	Computational Complexity	Storage Complexity
Alice	$O(\ 2^m\ )$	$O(2^m) * t$
Bob	$O(\ 2^{n'}\ )$	
Oscar	$O(\ 2^{m+n'}\ )$	

8.2.2. Variant of Merkle Cryptosystem (VMC)

Many variations of MC have been reported in literature. One of the variant (VMC), which is given in [55], uses larger key bit length for each puzzle. Also, puzzle generation method of VMC is different from MC. For this reason, it is not feasible for Bob to attack each puzzle similar to the original MC. This method uses another public

key X to generate public key vector such that length of the public key vector can be used to reduce search space of the participant of the communication. However, this method sends real index in clear text that significantly reduces brute force attack effort of Oscar. Parameters, which are used in VMC, are given below:

X : The public key value, which is used to generate public key vector P. Bit length of the single puzzle is equal to bit length of the secret key, $|P_i| = |K_i| = n$. Notice that, reasonable bit length of the $n \approx 64$ bits.

VMC algorithm is described below:

1. Alice generates puzzles $P_i = E_{K_i}(X)$ for $1 \leq i \leq \overline{N}$ where $\overline{N} = 2^m$. Alice sends vector P and public key X to Bob.

2. Bob generates random keys $l_1, l_2, ...$ and encrypts X with them. Bob compares results with elements in vector P such that there is a collision between encrypted value and one of the elements of vector P. Suppose that collision occurs for l_j. Consequently, $E_{l_j}(X) = P_{K_i}(X)$ and $l_j = K_i$. Bob finds the i'th puzzle in vector P via this collision search.

3. Bob sends index i in clear text together with encrypted message $M' = E_{K_i}(M)$. Bob sends (M', i) to Alice. Optionally, Bob might generate a session key K_s, $K_s' = E_{K_i}(K_s)$ and sends (K_s', i) to Alice.

4. Alice obtains index i and understands that Bob uses i'th key for secret communication. Alice decrypts message or session key $M = D_{K_i}(M')$ or $K_s = D_{K_i}(K_s')$.

Complexity of the VMC is summarized at Table 8.2.

Note that, VMC uses block ciphers to generate puzzles. The bit length of a single puzzle (n) is smaller than the key bit length of the block cipher such as AES having 128,192 or 256 bit key size in order to make collision search possible for Bob. In this

situation, first n bit of the block cipher key is used as variable part while remainder part is used as constant to obtain n bit security. In remainder of the study, n is used in this context.

Table 8.2. Computational and Storage Complexity of VMC

	Computational Complexity	Storage Complexity
Alice	$O(2^m)$	$O(2^m) * n$
Bob	$O(2^{n-m})$	
Oscar	$O(2^n)$	

8.2.3. Advantages of VMC over MC

Most important contribution of VMC is that the computational effort of Bob to find a secret key, is reduced from $O(\ 2^{n'}\ )$ to $O(2^{n-m})$. The reason is that, Bob realizes a collision search using advantages of large number of puzzles. In collision search, as explained in VMC step 2, Bob generates random keys and discovers corresponding private key with the probability of $Pr(Collision) = 2^m/2^n$. However, in original MC, number of puzzles does not give any contribution for reducing computational effort of Bob. The reason is that, Bob directly chooses one of the puzzles and realizes a brute force attack to the puzzle with $O(2^n)$ computational complexity.

VMC uses computational advantages of Bob to generate puzzles that have larger key bit length. Instead of giving Bob shorter time to determine a key, it is possible to keep collision search to constant but increase the bit length of single puzzle n. This approach increases computational effort of Oscar to break a single puzzle.

Suppose that Oscar knows the index Bob has found. In this case, in MC, the computational effort performed by Bob and Oscar would have been the same, which is in the order of $O(2^n)$. However, in VMC, even if Bob sends index in clear text, search space of Bob and Oscar are different. Bob does not choose a specific puzzle index but randomly discovers a corresponding private key with the probability of $Pr(Collision)$. Even if Oscar knows the index Bob has chosen, Oscar has to realize a brute force attack

with computational effort in the order of $O(2^n)$ while effort of the Bob is $O(2^{n-m})$. The reason is that, Oscar should find the same collision that Bob has found. The probability of Oscar to find the same collision as Bob is ignorable.

8.2.4. Disadvantages of VMC compared to MC

In VMC, Bob sends key agreement index i in clear text together with encrypted message as described in VMC step 3. Sending index in clear text causes significant security problem and drastically reduces the computational effort that Oscar has to perform, compared to the original MC. Notice that, in MC, Bob sends his selected index in clear text but the index sent by Bob does not correspond to the real index for the puzzle that Bob has found. Alice generates a puzzle in the MC step 1 such that it contains a pseudo-index which is known only by Alice. In MC Step 3, this pseudo-index is sent in clear text and does not reveal any information about the real index that Bob has found. Thus, brute force attack effort of Oscar is in the order of $O(2^{n+m})$. However, in VMC, sending real index i in clear text (Step 3) reduces the effort required to attack by Oscar from $O(2^{n+m})$ to $O(2^n)$. Note that the high number of puzzles, which is $\overline{N} = 2^m$, become useless to prevent attack of Oscar, since Oscar can observe real index and realizes brute force attack directly to the selected puzzle. Sending index in clear text may give small advantages compared to the original MC such that Alice does not make a search for pseudo-index. However, this search effort is completely insignificant since Alice stores pseudo-indices sorted and finds a corresponding real index easily. However, sending real index in clear text causes a significant security degradation that can not be compared with neglible search time advantage.

8.3. Improved Merkle Cryptosystem (IMC)

In this section, we present details of our Improved Merkle Cryptosystem (IMC). We make improvements over MC and VMC for three major points. Firstly, we increase security of the VMC by eliminating security problem which stems from sending real index in clear text. Notice that, computational advantages of Bob in VMC remain unchanged while security of the cryptosystem is increased. Secondly, we use auxiliary

secret keys, which increase security of the hashed secret value transmitted over network for key agreement. This approach provides security advantages over VMC for transmitted packets over network. Thirdly, we show that IMC reduces storage requirements and bandwidth consumption of Bob and Alice.

Following additional notations are used:

H : Cryptographic hash function. This hash function should be a secure hash function such as SHA family [28] or a cryptographic hash function having variable length output property such as HAVAL [29] (this may provide advantage for different bandwidth requirements of communication). y_i : Auxiliary secret key which is used to increase bit length security of the hashed message transmitted over network, P_i^* : Public key which is generated using y_i auxiliary keys. K_{s_a} and K_{s_b} denote session keys, which are generated by Alice and Bob, respectively. h : Secret hashed vector where $h_i \in h$, for all i, $1 \leq i \leq \overline{N}$ where $\overline{N} = 2^m$. $PRNG$: Pseudo Random Number Generator.

IMC algorithm is described below:

1.Alice generates auxiliary secret keys y_i and puzzle pairs $P_i = E_{K_i}(X)$, $P_i^* = E_{K_i}(y_i)$ for $1 \leq i \leq \overline{N}$ where $\overline{N} = 2^m$. Alice sends (P_i, P_i^*, X) for all i to Bob and stores (K_i, y_i) pairs as secret key pairs.

2. Alice generates hashed secret key vector h, $h_i = H(K_i||y_i)$ for $1 \leq i \leq \overline{N}$ where $\overline{N} = 2^m$. Alice stores h as secret key vector. She can store vector h in two different ways. Details for storage of vector h are given in Section 4.2.

3. Bob obtains (P_i, P_i^*, X) for $1 \leq i \leq \overline{N}$ where $\overline{N} = 2^m$. Then he generates random keys l_j similar to VMC step 2 such that $while(v, search\ on\ P_i)\{l_j = PRNG(), v = E_{l_j}(X), move\ indices\}$. If $(P_i == E_{l_j}(X))$ then $K_i = l_j$ and Bob finds one of the secret keys K_i. Using K_i, Bob decrypts P_i^* and obtains secret auxiliary key

$y_i = D_{K_i}(P_i^*)$.

4. Bob calculates $h' = H(K_i||y_i)$ and sends h' value to Alice. Notice that, only Alice knows K_i and y_i and using these secret key pairs, only Alice can calculate and verify h' value. Since one-way properties of H, Oscar can not find K_i and y_i from h'.

5. Session key agreement can be done with three different ways:

- *Alice decides session key:* Bob sends h' to Alice. Alice searches h' over vector h. If she finds then Alice and Bob agree on key $(K_i||y_i)$. Alice generates session key K_{s_s} and calculates $K_{s_s}{}' = E_{K_i||y_i}(K_{s_s})$ and sends $K_{s_s}{}'$ to Bob. Bob decrypts $K_{s_s}{}'$ and obtains $K_{s_s} = D_{K_i||y_i}(K_{s_s}{}')$.Alice and Bob agree on session key K_{s_s}.
- *Bob decides session key:* Bob generates K_{s_b} and calculates $K_{s_b}{}' = E_{K_i||y_i}(K_{s_b})$. Bob sends $(h', K_{s_b}{}')$ pair to Alice. Alice searches h' over vector h. If she finds then Alice and Bob agree on key $(K_i||y_i)$.Alice decrypts $K_{s_b}{}'$ and obtains $K_{s_b} = D_{K_i||y_i}(K_{s_b}{}')$.Alice and Bob agree on session key K_{s_b}.
- *Alice and Bob jointly decide session key:* Alice and Bob agree on $(K_i||y_i)$ similar to steps above and they exchange K_{s_s} and K_{s_b} session keys. They calculate their joint session key $K_s{}' = K_{s_s} \oplus K_{s_b}$.

8.4. Analysis and Comparison of IMC

In this section, we analyze properties of IMC and compare it to the MC and VMC. Also, we compare the security of IMC to the some well-known public key cryptosystems. Firstly, we analyze security properties and advantages of IMC over MC and VMC showing that IMC provides higher security than MC and VMC. Secondly, we present storage advantages of IMC over MC and mention some additional techniques to reduce storage requirements of IMC. Table 8.3 summarizes the resulting improvements and comparisons of IMC to the MC and VMC. Computational and storage complexity of each method are expressed in terms of the parameters m, n, n', t for Alice, Bob and Oscar. Table 4 demonstrates comparison of the IMC to MC, VMC and some other public key cryptosystems for their bit length security measurement regarding to various

criteria.

8.4.1. Security Analysis and Advantages of IMC

In IMC, in order to hide key agreement value, which is secret key $(K_i||y_i)$, we calculate hash of $(K_i||y_i)$, $h' = H((K_i||y_i)\ )$ and transmit h' over network. Due to one-way property of cryptographic hash functions, Oscar can not find $(K_i||y_i)$ from h'. With our improvement, in order to obtain $(K_i||y_i)$, Oscar has to realize brute force attack to all puzzles ($\overline{N} = 2^m$ puzzles). Since brute force attack to a single puzzle requires $O(2^n)$ computational effort, total computational effort of Oscar becomes $O(2^{n+m})$. In VMC, since real index is sent in clear text, Oscar knows which index Bob has chosen. Thus, computational effort of Oscar in VMC is only $O(2^n)$. In table 8.3, we can see security advantages of the IMC over VMC ($O(2^{n+m}) > O(2^n)\ $).

IMC can use larger key bit length for a single puzzle by shifting computational advantages of Bob to the overall system security (properties of VMC). Shifting computational advantages of Bob to the key bit length of a single puzzle (parameter for overall system security), we can select n such that $n > n'$. Thus, $O(2^{n+m}) > O(2^{n'+m})$ and IMC can provide higher security than MC using this approach. In table 8.3, advantages of IMC over MC can be seen. In addition to this, in table 8.3, security/performance advantage of IMC over MC and VMC is shown. It is calculated by dividing computational effort of Oscar to the computational effort of Bob. This gives us a criterion about the efficiency of the cryptosystem. We can see that both MC and VMC have $O(2^m)$ security/performance value while IMC has $O(2^{n+m})/O(2^{n-m}) = O(2^{2m})$ which is more efficient than MC and VMC.

Another improvement of IMC is that it uses auxiliary key y_i to increase bit length security of the hashed key agreement value h'. Suppose that Oscar obtain h' value by eavesdropping. In order to find $(K_i||y_i)$ from h', Oscar should try all possible $O(2^{2n})$ key space for detecting a one-to-one mapping among generated random keys and h' value. One-way properties of cryptographic hash function does not allow Oscar to recover $(K_i||y_i)$ from h' without brute force attack under the assumption of random behavior

of hash functions [2]. Notice that, $|(K_i||y_i)| = 2n$ and for $n \simeq 70$ bits, $|(K_i||y_i)| = 140$ bits. This provides the security in the order of $O(2^{140})$. If only $h' = H(K_i)$ was used instead of $h' = H(K_i||y_i)$ then brute force effort of Oscar would have been $O(2^n)$. Under this condition, security of the transmitted message over network ($O(2^n)$) would have been lower than security of overall system cryptosystem ($O(2^{n+m})$) and Oscar would have broken system easily by attacking h' value instead of P_i puzzles. The main idea behind of the using auxiliary keys is preventing IMC from this attack. In VMC, messages transmitted over network are encrypted using only n bit K_i keys. Thus, IMC provides higher security for messages transmitted over network (including key agreement value) than that of the VMC. In MC, session keys are embedded into puzzle P_i. When Bob solves the puzzle, he uses session key to encrypt message, which is transmitted over network. Thus, message security of MC depends on key bit length of the session key and overall security of the cryptosystem. Results are summarized at Table 8.3.

8.4.2. Storage Analysis and Advantages of IMC

IMC has storage advantages over MC. In MC, a single puzzle P_i contains three components, which are S, $\overline{r}_i$, and K_{s_i}, respectively (total t bits). These additional components increase bit length of a single puzzle and cause significant storage and transmission load. However, in IMC, there are puzzle pairs (P_i, P_i^*) each of them having $2n$ bit length. Thus, for $\overline{N} = 2^m$ puzzles, IMC provides $O(2^m t - 2^{m+1} n) = O(2^m(t-2n))$ storage advantages over MC. For example, bit length of a single puzzle in MC with 40 bit redundancy, 40 bit pseudo-index and 128 bit session key are approximately $t \approx 208$ bit. In IMC, the bit length of a key can be selected up to 70 bits (due to storage and computational limits). Thus, bit length of a single puzzle pair is $2n \approx 140\ bits$. Consequently, for $m \approx 30$, IMC provides storage advantages up to $(2^{30} * 68) \simeq 1\ GB$ for these settings when compared to MC. Important point is that, same amount of gain is also obtained for network bandwidth consumption. Notice that, VMC has a small storage advantages when compared to IMC (IMC : $O(2 * 2^m n)$, VMC: $O(2^m n)$). However, for corresponding small storage load, IMC has significant security advantages over VMC. These results can also be observed in table 8.3.

Apart from these, in IMC step 2, we have discussed that secret key vector h can be stored in two different ways. This is a tradeoff approach among storage and computational resources of Alice. If Alice has sufficient storage resources, she stores vector h permanently. Then, whenever a key agreement occurs, Alice directly searches h' over vector h for key agreement. This approach provides computational resource advantage. However, if Alice does not have sufficient storage capability, for each key agreement, she dynamically generates h_i elements using (K_i, y_i) secret key pairs and compares h_i with h' to find a match. Thus, Alice does not have to store vector h permanently. Since cryptographic hash functions are fast, with feasible amount of puzzle ($\overline{N} = 2^m, m \approx 30$), search operation becomes feasible. This approach provides storage advantage.

Table 8.3. Comparison of IMC to MC and VMC

		MC	VMC	IMC
Computational Complexity	Alice	2^m	2^m	2^m
	Bob	$2^{n'}$	2^{n-m}	2^{n-m}
	Oscar	$2^{n'+m}$	2^n	2^{n+m}
Storage Complexity	Alice	$2^m t$	$2^m n$	$2^{m+1} n$
	Bob	$2^m t \rightarrow 1$	$2^m n \rightarrow 1$	$2^{m+1} n \rightarrow 1$
	Oscar	$2^m t$	$2^m n$	$2^{m+1} n$
Security Comparison		$2^{n'+m}$	2^n	2^{n+m}
Security/Computational		2^m	2^m	2^{2m}
Message Security		$\lvert K_s \rvert$	2^n	2^{2n}

8.4.3. Comparison of IMC with MC, VMC and Some Well-known Public Key Cryptosystems

Table 8.4 demonstrates comparison of the IMC with MC, VMC and some well-known public key cryptosystems. Symmetric Cryptosystem Bit Length (SCBL) security gives total bit length strength of the MC, VMC and IMC to resist attack of Oscar. For example, 100 bits mean that computational effort of Oscar to break cryptosystem is equivalent to break 100 bits block cipher. Note that, it does not mean that bit length of the key that will be used for block cipher is 100 bits, but total effort (using

all puzzles in the system) corresponds to 100 bits security. To reach this security level, parameters $m \simeq 30$ bits, $n' = 40$ bits and $n = 70$ bits are selected for today's and near future feasible memory and computational possibilities. Brute force attack capability of Bob is selected as 2^{40} that allows feasible search time for key agreement. Storage capability of Alice and Bob is selected as approximately $2^{30} \cdot 140$ bits so that it is feasible for current hardware possibilities. Using these parameters, maximum security available for the MC is 2^{70}. In IMC, using aforementioned improvements, security level can be reached up to 2^{100} bits ($O(2^{n+m})$) that extends approximate lifespan of the cryptosystem to 30 years (Table 4) [77]. For these parameters, providing more than 70 bit security becomes infeasible both for MC and VMC. Remainder parts of the table 8.4 shows equivalent bit length security level for various public key cryptosystems and their related lifespan and economical cost values. Corresponding values for symmetric key bit length security are obtained from [77]. For these comparisons, [79] can also be used. With these interpretations, we see that IMC can provide as high security as some well-known public key cryptosystems. In table 8.4, following abbreviations are used: $PKCL$: Public Key Cryptography bit Length. CAS: Classical Asymmetric Cryptography like RSA. $SDLF$: Sub Group Discrete Logarithm problem Field. EC: Elliptic Curve. LB: Lower Bound.

Table 8.4. Comparison of IMC with VMC-MC and some well-known public key cryptosystems for various criteria

SCBL	MC	70	Infeasible for Participants				
	VMC	70	Infeasible for Participants				
	IMC	70	76	82	88	94	100
PKCL	CAS	952	1279	1613	2054	2560	3137
	SDLF	704	960	1248	1632	2080	2592
	SDL Key Size	125	135	145	156	167	178
	EC Size	132	155	173	197	218	240
Infeasible Number of MIPS Years		$8 \cdot 10^9$	$5 \cdot 10^{11}$	$2 \cdot 10^{13}$	$2 \cdot 10^{15}$	$1 \cdot 10^{17}$	$8 \cdot 10^{18}$
LB for HW attack cost for 1 day breaking		$1 \cdot 10^8$	$3 \cdot 10^8$	$4 \cdot 10^8$	$7 \cdot 10^8$	$1 \cdot 10^9$	$2 \cdot 10^9$
Corresponding Lifespan		2000	2008	2015	2023	2031	2039

8.5. Conclusions and Future Works

In this study, we propose Improved Merkle Cryptosystem (IMC), which can be considered as an alternative method for key agreement schemes, based on only symmetric cryptosystem and cryptographic hash functions without requiring a Trusted Third Part (TTP). As a novelty, IMC uses cryptographic hash functions and auxiliary keys to increase security of MC and VMC. Unlike VMC, IMC hides key agreement value using cryptographic hash functions and enhances the security of key agreement value utilizing auxiliary keys. These approaches provide significant security advantages over VMC. Since IMC utilizes some advantages of VMC over MC, IMC also provides higher security than MC. Different puzzle structure of IMC reduces storage requirement of the cryptosystem when compared to MC. Our improvements provide a solution to use MC for long term security, which is compatible with some well-known public key cryptosystems, within today's feasible hardware possibilities.

MC does not provide security against active attacks such as message replay and injection attacks. As a future work, we consider using IMC to develop a key agreement scheme, which can provide major cryptographic goals such as confidentiality, integrity, authentication and unforgeability together. In order to this, we consider using some principles of signcryption [20]. We will integrate IMC with a signcryption based key exchange schemes [30], which uses nonce and time-stamps to prevent cryptosystem from active attacks. We believe that, this integrated cryptosystem, Signcryption Type Authentic Key Establishment scheme (STAKE), will solve active attack problems of IMC and will provide additional cryptographic goals.

9. CONCLUSIONS

Wireless communication networks become more prevalent and important in today's modern communication systems requiring high security and performance together. However, the naturally broadcast structure and energy, bandwidth, storage and computational resource limited characteristics of wireless communication networks make them very challenging networks for achieving high security and performance together. Similar to other network systems, wireless networks need to achieve major cryptographic goals such as confidentiality, authentication, integrity, unforgeability and non-repudiation. Moreover, wireless networks are especially vulnerable to some active attacks such as message replay and insertion, which require additional cryptographic precautions. In addition to these, wireless communication networks, taken into consideration their broadcast nature, are generally used for group communication and especially for data multicast applications. Notice that, secure group communication requires two additional security services, forward and backward security in order to provide freshness of the cryptographic keys in the group. In this sense, secure group communication requires complex key management techniques in order to achieve major cryptographic goals, preventing from specific active attacks and providing forward and backward security. Forward and backward security services also need major cryptographic services working together. Thus, providing forward and backward security in a very large and dynamic wireless networks is a very challenging issue.

In this thesis, taken into consideration these problems, we proposed novel wireless network security mechanisms and cryptographic methods each of them provides different and efficient solutions for various wireless network types, structure and security-performance requirements. We especially worked on secure satellite multicast security mechanisms and military MANET security mechanisms. Satellite multicast systems, covering large areas of the world, include very large number of members having dynamic join-leave characteristic. Also, satellite networks suffer from severe delay and packet loss problems. Thus, we designed our satellite security mechanisms so that they can achieve aforementioned security goals under these conditions efficiently. Notice

that, military MANETs have more severe conditions in hostile environments and more resource limited possibilities. In addition to this, they need very high security and performance together. We specifically designed a military MANET security mechanism taking into consideration these problems.

Our proposals are based on three major approaches: Structural design, hybrid key management techniques and novel cryptographic methods. Using these three major approaches, we designed new security mechanisms for various wireless network structures and security requirement. Structural design and hybrid key management techniques are used in integrated manner. Our main structural design concept is the multi-tiered structure, which is integrated with centralized and decentralized key management techniques. In this way, we can scale very large and complex networks with much smaller cost than traditional approaches. Each tier is managed by an appropriate centralized key management protocol and sub-groups in the tiers are controlled by a decentralized key management protocol. These approaches firstly eliminate single point of failure problem that exist in pure centralized key management approaches. Also, they utilize logarithmical scale properties of centralized key management protocols in each tier and sub-groups. In this way, moderately large and dynamic sub-groups can be handled with a smaller cost. Notice that, these approaches require highly efficient and secure cryptographic methods in order to provide the desired security level.

In order to fulfill these requirements, we utilized and adapted various cryptographic methods to our security mechanisms. These cryptographic methods are selected so that they can provide high security while minimizing bandwidth consumption. The reason is that, if appropriate cryptographic methods are not used then our multi-tier approaches may cause delay problems (multiple encryption among tiers). Thus, we essentially used hybrid cryptography techniques, which are used to transmit required GK and KEKs to members securely. Then, bulk data multicast is done with symmetric cryptography using these keys that minimizes delay problems. In our mechanisms, as novel approaches, we utilized message recovery type ECC based digital signature, specific authentic key establishment schemes and signcryption based cryptographic techniques, which have many advantages over traditional cryptographic

approaches. In traditional methods, classical DH, ECDH and GAP-DH related methods are used. Notice that, DH, ECDH and GAP-DH methods, if they are implemented without specific authentication mechanisms, can not provide the desired level of security. Also, GAP-DH based methods are computationally expensive. In order to achieve major cryptographic goals, digital signature based approaches are needed. However, especially for small messages, these methods cause significant message overhead and create unnecessary bandwidth consumption. In this sense, our cryptographic methods have security, performance, storage and bandwidth consumption advantages when compared to these methods.

We proposed TTPVSS, which uses two independent LKH tier and ECPVSS as a major cryptographic method. Two independent LKH tier provide independency of tiers principle. Whenever a member join-leave event occurs, effects of the event are encapsulated in the related part of the SSMS and other parts of the system are not affected from modifications. Batch keying mechanism, using group key seeds, provides additional bandwidth advantage. TTPVSS assumes the availability of TUs in SSMS. Notice that, this assumption is compatible with today's modern SSMSs that already work in this sense. Satellite-TU and TU-members tiers work in integrated manner without creating massive workload to each other. Using ECPVSS in SSMS is a novel approach. MRDS properties of ECPVSS are one of the most efficient choices to transmit KEKs in the SSMS. Also, random characteristic of cryptographic keys make ECPVSS more secure for TTPVSS. ECPVSS provides at least three times better bandwidth usage when compared to traditional approaches. Thus, we can see that, two independent LKH tiered structure is a useful approach for SSMS. Also, using ECC based MRDS cryptographic methods are especially useful and promising approaches.

We proposed a new three-tier satellite multicast security mechanism based on ECMQV and IMC. This security mechanism utilizes independency of tiers principle of TTPVSS. However, our three-tier security mechanism uses a different structure to scale and to secure large SSMS. Firstly, this mechanism uses properties of GEO, MEO and LEO satellites in order to minimize delay and packet loss problems. Also, availability of extensive satellite internetworking helps batch keying mechanism. GEO satellites,

covering large areas of the world and having better computational possibilities, generate necessary cryptographic keys, seeds and tickets and distribute them using ECMQV to LEO and MEO satellites. Notice that, since LEO-MEO satellites and GEO satellites can be accepted as both secure and nearly equal position for cryptographic key generation rights, we have preferred a key exchange algorithm instead of key transport method such as ECPVSS. LEO-MEO satellites use the advantages of lower delay rate and realize data multicast. Also, this security mechanism uses additional components such as NTS in order to facilitate distribution of the public keys and certificates.

We proposed NAMEPS, which uses N-tier independent ELK tier and multi-recipient signcryption schemes. NAMEPS also utilizes principles of TTPVSS and our three-tier satellite multicast security mechanism but uses a different structure, key management protocol and cryptographic method. Using ELK provides efficient member join possibilities. Using validation ticket mechanisms reduces roaming workload of the MMUs. Using Multi-recipient signcryption scheme is a novel approach and brings all advantages of signcryption schemes to NAMEPS and also compatible with multicast nature of SSMSs.

We proposed HIMUTSIS, which is specifically designed for mission critic military MANETs. HIMUTSIS brings many novelties for structural design approaches of military MANETs. These approaches minimize SPoF problem, rekeying workload in military MANETs and threshold cryptography requirements. HIMUTSIS also presents multi-level security structure and provides different choices for different tiers of military MANETs. In addition to these, as a novel approach, HIMUTSIS adapts SDSS-1 signcryption based DKEUTS protocol to multi-tiered multi-security level military MANET structure, which provides performance and security advantages.

We proposed IMC, which significantly improves MC and VMC for both security and performance aspects. IMC utilizes collision search approach of VMC but improves it by hiding key agreement value using a cryptographic hash function. In order to increase security of the MC and VMC, IMC also uses a different puzzle structure, which provides advantages when compared to MC. With these improvements, IMC

provides as high security as today's modern public key cryptosystems while VMC and MC can not provide the same level of security. In addition to IMC, we work on STAKE that combines recoverable commitment value concept of the signcryption schemes and principles of IMC to create a key establishment protocol that is resistant for both active and passive attacks.

As a result, we can say that, in order to provide efficient and secure solutions to most challenging problems of wireless communication networks, co-operative approaches are needed which focus on various aspects of the security system such as efficient structural design, original hybrid key management techniques and novel cryptographic approaches. Integration of all these methods can provide efficient solutions for wireless security networks. In this thesis, with seven studies, we showed that, these integrated approaches provide highly efficient solutions to SSMS and military MANETS. However, we also mentioned that our generic design principles can be extended to all wired and wireless networks. We see that, independency of tiers principle with hybrid key management techniques is a key concept of our design that provides major performance gain. As cryptographic methods, ECC based MRDS and ECC based authentic key exchange schemes are highly useful for wireless networks. In addition to these, we believe that, utilizing signcryption based cryptographic approaches to wireless network security is a promising approach.

As future works, we will extend our major design principles to other wireless communication networks such as wireless sensor networks. Also, we want to focus on providing security in very large and integrated digital battle field communication networks that include satellites, military MANETs and sensor networks together. We will work on other efficient signcryption based schemes such as identity based and group based signcryption schemes to provide security in these networks.

REFERENCES

1. A. Altay Yavuz, F. Alagoz and E. Anarim, " A new protocol for satellite multicast security" , Fifth GAP. Engineering Congress, Sanliurfa, Turkey, April 2006.

2. D. Stinson. Cryptography Theory and Practice. CRC Press, Inc., Third Edition, 2005.

3. A. Menezes, P. Van Oorschot and S. Vanstone. Handbook of applied cryptography. CRC press, 1996.

4. Kim, Y., Perrig A. and Tsudik G., " Simple and fault-tolerant key agreement for dynamic collaborative groups" , In Proceedings of the 7th ACM Conference in Computer and Communication Security, (Athens, Greece Nov.). (S. Jajodia and P. Samarati, Eds.), pp. 235–241, 2000.

5. D. H. S. Rafaeli, " A survey of key management for secure group communications" , ACM Comp. Surveys, vol. 35, no. 3, Sept 2003, pp. 309–29.

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. Wallner, E. Harder and R. Agee, " Key management for multicast: Issues and architectures" , IETF, RFC2627, June 1999.

8. D. Balenson, Alan T. Sherman and David A. McGrew, " 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.

9. 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.

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

11. S. Setia, S. Koussih and S. Jajodia, " Kronos: A scalable group re-keying approach for secure multicast" , In Proceedings of the IEEE Symposium on Research in Security and Privacy, May 2000.

12. 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).

13. A. Altay Yavuz, F. Alagoz and E. Anarim, " Three-Tiers satellite multicast security protocol based on ECMQV and IMC methods" , 11th Computer-Aided Modeling, Analysis and Design of Communication Links and Networks(CAMAD'06), June 2006.

14. H. Harney, C. Muckenhirn and T. Rivers, " Group Key Management Protocol (GKMP) Architecture" , Request for Comments (RFC) 2093, Internet Eng. Task Force, July 1997.

15. Kaveh Pahlavan and Prashant Krishnamurthy, Principles of Wireless Networks A unified Approach, Pearson Education, Prentice Hall PTR, 2002.

16. A. Altay Yavuz, F. Alagoz and E. Anarim, " A new satellite multicast security protocol based on elliptic curve signatures" , 2nd IEEE International Conference on Information & Communication Technologies(ICTTA) , April 2006.

17. Wanldvogel M., Caronni G., Sun D., Weiler N. and Plattner B., " The VersaKey framework: Versatile group key management" , IEEE J. Sel. Areas Commun. (Special Issue on Middleware) 17, 9 (Aug.), 1614–1631, 1999.

18. A. T. Sherman and D. A. McGrew, " Key establishment in large dynamic groups using one-way function trees" , IEEE Transactions on Software Engineering, vol.

29, no. 5, pp. 444–458, 2003.

19. Oded G., Shafi G. and Silvio M., " How to construct random functions" , Journal of the ACM, 33(4):792–807, October 1986.

20. Y. Zheng, " Digital signcryption or how to achieve Cost(Signature Encryption) $\ll$ Cost(Signature) + Cost(Encryption)" , Advances in Cryptology, Crypto'97, Lecture Notes in Computer Science, Vol. 1294, pp. 165-179, Springer-Verlag, 1997.

21. Y. Zheng, " Signcryption and its applications in efficient public key solutions" , Proceedings of 1997 Information Security Workshop (ISW'97), Lecture Notes in Computer Science, vol.1397, pp.291-312, Springer-Verlag, 1998.

22. Y. Zheng and H. Imai, " Compact and unforgeable key establishment over an ATM network" , Proceedings of IEEE INFOCOM'98 , pp.411-418, 29/3-3/4, 1998.

23. A. Altay Yavuz, F. Alagoz and E. Anarim, " Improved Merkle Cryptosystem (IMC)" , Submitted to ISCIS'06, November 2006, Istanbul, Turkey.

24. 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.

25. L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, " An efficient protocol for authenticated key agreement" Designs, Codes and Cryptography, pages 28:119–134, 2003.

26. H. Krawczyk, HMQV, " A high-performance secure Diffie-Hellman protocol" , Advances in Cryptology, CRYPTO 2005, Lecture Notes in Computer Science, 3621 (2005), 546-566.

27. T. ElGamal, " A public key cryptosystem and a signature scheme based on discrete logarithms" , IEEE Transactions on Information Theory, IT-31(4):469-472, 1985.

28. NIST. Secure Hash Standard. Federal Information Processing Standards Publications(FIPS PUB) 180-2, August 26, 2002. U.S. Department of Commerce, N.I.S.T.

29. Y. Zheng and J. A. Pieprzyk, " Haval - a one-way hashing algorithm with variable length of output" , In J. Seberry and Y. Zheng, editors, Advances in Cryptology - AUSCRYPT'92, volume 718 of Lecture Notes in Computer Science, pages 83-104. Springer-Verlag, 1993.

30. Y. Zheng, " Shortened digital signature, signcryption, and compact and unforgeable key agreement schemes" , (A contribution to IEEE P1363 Standard for Public Key Cryptography), July 1998.

31. W. Mao. Modern Cryptography Theory & Practice. Hewlett-Packard Company, Prentice Hall, 2004

32. P. Horster, M. Michels and H. Petersen, " Meta-ElGamal signature schemes" , In Proceedings of the second ACM Conference on Computer and Communications Security, pages 96-107, New York, November 1994. The Association for Computing Machinery.

33. Y. Zheng and H. Imai, " How to construct efficient signcryption schemes on elliptic curves " , Information Processing Letters, Vol.68, pp.227-233, 1998.

34. W. Mao and J. Malone-Lee, " Two birds one stone: Signcryption using RSA " , In Marc Joye, editor, Progres in Cryptology — CT-RSA 2003, Lecture Notes in Computer Science. Springer-Verlag, 13–17 April 2003.

35. J. Baek, R. Steinfeld and Y. Zheng, " Formal Proofs for the security of signcryption" , In D. Naccache and P. Pallier, editors, Public Key Cryptography 2002(PKC 2002), volume 2274 of Lecture Notes in Computer Science, pages 80-98. Springer-Verlag,1998.

36. W. Diffie and M. E. Hellman, " New Directions in Cryptography " , IEEE Trans.

Information Theory, vol. IT-22, Nov. 1976, pp: 644 654.

37. Don Johnson and Alfred Menezes, " The Elliptic Curve Digital Signature Algorithm(ECDSA)", February 24, 2000.

38. Alexander W. Dent, " Hybrid cryptography" , Cryptology ePrint Archive, Report 2004/210, 2004. http://eprint.iacr.org/2004/210/.

39. M. P. Howard, S. Iyengar, Z. Sun and H. Cruischank, " Dynamics of key management in secure satellite multicast" IEEE Journal on Selected Areas in Communications, Vol. 22, No.3, Feb 2004.

40. 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.

41. R. L. Rivest, A. Shamir and L. Adleman, " A method for obtaining digital signatures and public key cryptosystems" , Communications of the ACM, 21(1978), 120-126.

42. Standard specifications for public key cryptography. IEEE P1363/D13, November 1999.

43. M.Steiner, G. Tsudik and M. Waidner, "Diffie-Hellman Key Distribution Extended to Groups", Proc. 3rd ACM Symp. on Computer and Communications Security, Vol. 1, pp31-37, March 1996.

44. Kristin Lauter, " The Advantages of elliptic curve cryptography for wireless security" IEEE Wireless Communications, February 2004.

45. NIST. Specifications for the Advanced Encryption Standard(AES). Federal Information Processing Standards Publications (FIPS PUB) 197, November 2001. U.S. Department of Commerce, N.I.S.T.

46. B. Schneier,. Applied Cryptography. New York: Wiley, 1996.

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

48. 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.

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

50. N. Smart and P. Leadbitter, " Analysis of the insecurity of the ECMQV with partially known nonces" In Proceedings ISC 2003, pages 240–251. Springer-Verlag LNCS 2851, August 2003.

51. M. Adriano Strangio, " Efficient Diffie-Hellmann two-party key agreement protocols based on elliptic curves" , Proceedings of the 2005 ACM symposium on Applied computing, Pages: 324 - 331, 2005.

52. A. Menezes, Another look at HMQV, In http://eprint.iacr.org/2005/205, June 27, 2005.

53. Lenore Blum, Manuel Blum and Michael Shub, " A simple unpredictable pseudo-random number generator" , SIAM Journal on Computing, volume 15, pages 364–383, May 1986.

54. C. Merkle, " Secure communications over insecure channels" , Communications of the ACM 21(4), pp294–299 (April 1978).

55. Chris Mitchell, Public key encryption using block ciphers, technical report RHUL-MA-2003-6, 9 September.

56. C. Chen, E. Ekici and I.F. Akyildiz, " Satellite grouping and routing protocol for

LEO/MEO satellite networks " , Proceedings of the Fifth ACM WoWMoM 2002, September 28, 2002, pp. 109-116.

57. W. Werner, A. Jahn, E. Lutz and A. Bottcher, " Analysis of System Parameters for LEO/ICO-Satellite Communication Networks" , IEEE Journal on Selected Areas in Communication, 13(2):371-381, February.

58. Rhee, Y. Park and G. Tsudik, " A group key management architecture in mobile ad-hoc wireless networks " , Journal Of Communication and Networks, Vol. 6, No. 2, pp. 156-162, June 2004.

59. A. Altay Yavuz, F. Alagoz and E. Anarim, " HIMUTSIS: Hierarchical Multi-Tier Adaptive Ad-hoc Network security protocol Based on Signcryption Type Key Exchance Schemes " , Submitted to ISCIS'06, November, 2006, Istanbul, Turkey.

60. D. L. Gu, G. Pei, H. Ly, M. Gerla and X. Hong, " Hierarchical Routing for Multilayer Ad-hoc Wireless Networks with UAVs" , In IEEE MILCOM, 2000.

61. J. Kong, H. Luo, K. Xu, D. Lihui Gu, M. Gerla and S. Lu, "Adaptive Security for Multi-layer Ad Hoc Networks" , Wireless Communications and Mobile Computing, Special Issue on Mobile Ad Hoc Networking, vol. 2, pp. 533– 547, 2002.

62. N. Asokan and P. Ginzboorg, " Key Agreement in Ad-hoc Networks" , In Computer Communications, 23(18), pp. 1627-1637, 2000.

63. L.Zhou and Z. Hass, " Securing ad hoc networks" , IEEE Network, 13(6), pages 24-30, November/December 1999.

64. Gang Yao, Kui Ren, Feng Bao, Robert Deng and Dengguo Feng, "Making the Key Agreement Protocol in Mobile Ad Hoc Network More Efficient" , In Proc. of ACNS 2003,LNCS, Vol. 2846, p343-356, 2003.

65. D. Augot, R. Bhaskar and V. Issarny and D. Sacchetti, " An Efficient Group Key Agreement Protocol for Ad hoc Networks" , IEEE Workshop on Trust, Security

and Privacy in Ubiquitous Computing, Taormina, Italy, 2005.

66. D. L. Gu, G. Pei, H. Ly, M. Gerla, B. Zhang and X. Hong, " UAV-aided Intelligent Routing for Ad-hoc Wireless Network in Single-area Theater" , In IEEE WCNC, pages 1220–1225, 2000.

67. A. Klimov and A. Shamir, " New Cryptographic Primitives Supported on Multi-word T-Functions" , In B. Roy and W. Meier, editors, Fast Software Encryption 2004, volume 3017 of LNCS, pages 15. Springer, 2004.

68. Vladimir Anashin , Andrey Bogdanov, Ilya Kizhvatov. ABC : A New Flexible Stream Cipher, available at http://www.ecrypt.eu.org/stream/abc.html.

69. A. Klimov and A. Shamir, " Cryptographic Applications of T-functions" , Selected Areas in Cryptography (SAC), 2003.

70. A. Altay Yavuz, F. Alagoz and E. Anarim, " NAMEPS: N-tier sAtellite Multicast security Protocol based on multi-recipient Signcryption schemes (NAMEPS)" , To appear IEEE 2006 Global Communications Conference (GLOBECOM 2006), San Francisco, CA 2006.

71. Ueli Maurer, " Cryptography 2000 -10 Years Back, 10 Years Ahead" , Lecture Notes in Computer Science, Springer-Verlag, vol. 2000, pp.63-85, 2001.

72. J. Hoffstein, J. Pipher and J.H. Silverman, " NTRU: A Ring-Based Public Key Cryptosystem" , Proceedings of ANTS III, Portland, June 1998.

73. R. L. Rivest, A. Shamir and D. A. Wagner, Time-lock puzzles and timed-release crypto, MIT LCS Tech. Report MIT/LCS/TR-684, 1996.

74. Aldar C., F. Chan and Ian F. Blake, " Scalable, Server-Passive, User-Anonymous Timed Release Cryptography" icdcs, pp. 504-513, 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05), 2005.

75. B. Bencsáth, L. Buttyán and I. Vajda, “ A game based analysis of the client puzzle approach to defend against DoS attacks” , Proceedings of SoftCOM 2003. 11. International conference on software, telecommunications and computer networks, University of Split, pp. 763-767.

76. D. Dean and A. Stubblefield, “ Using client puzzles to protect TLS, Proceedings of the USENIX Security Symposium” , August 2001.

77. Arjen K. Lenstra and Eric R. Verheul, “ Selecting cryptographic key sizes” , Journal of Cryptology, 14(4):255–293, 2001.

78. A. Altay Yavuz, E. Anarim and F. Alagoz, “ IMC (Improved Merkle Cryptosystem and STAKE (Signcryption Type Authentic Key Establishment Scheme) ” , under preparation, study will be submitted to “ International Journal of Computational Intelligence (IJCI)” , 2006.

79. A.K. Lenstra, “ Unbelievable security, Proceedings Asiacrypt 2001 ” , LNCS 2248, Springer-Verlag 2001, 67-86.