SURVEY AND TAXONOMY OF KEY MANAGEMENT PROTOCOLS FOR WIRED AND WIRELESS NETWORKS

Adarsh Kumar[1], Alok Aggarwal[2] and Charu[3]

[123]Department of Computer Science and Information Technology, Jaypee Institute of Information and Technology, Noida, INDIA
[1]adarsh.kumar@jiit.ac.in, [2]alok.aggarwal@jiit.ac.in,
[3]charu.kumar@jiit.ac.in

ABSTRACT

The purpose of this paper is to survey the key management protocols for wired and wireless networks and study their security aspects in terms of key generation, agreement and distribution. The central research challenge is exhaustive survey of secure and efficient key management protocols. In this survey, it is shown that all these protocols could be placed under one of two key management protocol categories: (i) peer to peer communication and (ii) group communication. This can also be analyzed that peer to peer key management can be classified as: (i) symmetric key, (ii) asymmetric key and (iii) hybrid key management protocols and group communication can further be classified as: (i) Diffie-Hellman based (ii)Hybrid key management. We can say that our theoretical and execution analysis of protocols emphasise various observations that can motivate researchers in key management issues of networks.

KEYWORDS

Key Management Protocols, Peer to Peer, Group, Multicast, Hierarchical, Tree based, Trusted Third Party, Escrow less, Server based, Server less

1. INTRODUCTION

Cryptography is the study of various schemes used for encryption of data for secure communication. These schemes are collectively known as cipher or cryptographic systems. These ciphers are characterized by:-

(i) The type of operations used for transforming plaintext to ciphertext. For example, substitution and transposition.

(ii) Processing methodology for plaintext. For example, Block processing or stream processing.

(iii) Number of keys used to convert plaintext to ciphertext. For example, symmetric/single or asymmetric/two key/public key.

Key is an element which can be either numeric or non-numeric, which when applied to a given message results in a encrypted message. Key can be implicitly or explicitly derived from plaintext. Implicit key derivation is also known as auto keying, where the derived key is a part of the plaintext. Explicit key or individual key is a key that is not a part of the plaintext. For a secure communication to take place, the life cycle of key involves: initialization, agreement, distribution and cancellation. This entire process is also known as key management. Figure 1 show the life cycle of key management.

DOI : 10.5121/ijnsa.2012.4302

This paper is organized as follows: on 2 discusses the related work. Section 3 presents the key initialization classifications and procedures. In section 4 details execution on symmetric and asymmetric protocols has been done. It also discusses the strengths and weaknesses of the protocols. Section 5 discusses the key cancellation classifications and procedures. In section 6 conclusions are drawn based on the survey. Lastly section 7 explains the future direction of this work.

2. RELATED WORK

In earlier days, the information was stored on papers, magnetic tapes, floppy disks, compact disks, USB flash drives, external hard disks etc. This stored information was distributed through courier services-informally called sneaker net. Disadvantage of this type of mechanism was that there was need of a very high security environment to be constructed, where there was need of strong manual exercise. Constructing such a highly secured environment manually is nearly impossible. Now a days, various protocols/techniques have been defines to securely transmit the private/public part of key to destination [1, 2, 3, 4, 5]. Some of the key management mechanisms are: (i) File-Based Protocol (ii) Mail Based Protocol (iii) HTTP/HTTPS-Based Protocol (iv) TCP-Based Protocol. Key management can be classified as: key agreement and key distribution. This classification is based on type of cryptography i.e. symmetric and asymmetric. Various symmetric key agreement protocols are: Boyd's key agreement protocol, Bluetooth key agreement, ISO/IEC 11770-2 protocols etc. Symmetric / private key distribution protocols can be classified as: (i) Trusted Third Party (TTP) based protocols, (ii) Without TTP (iii) Contributory. Some of the symmetric Key Cryptography based protocols are: (i) lolus, (ii) logical key hierarchies and Key Graph, (vi) Kerberos etc. Various key agreement mechanisms [6] in asymmetric key/public key cryptosystem technique are: (i) Internet Security Association and Key Management Protocol (ISAKMP)[7] (ii) Oakley Key Determination Protocol (OKDP)[8], (iii) SKEME: A Versatile Secure Key Exchange Mechanism for Internet[9] etc. Asymmetric key distribution protocols are: Blake-Wilson-Menezes (BMW), Needham Schroeder public key etc. Another classification is based on: (i) Peer to peer communication and (ii) Group communication. Figure 1 shows the classification of two categories.

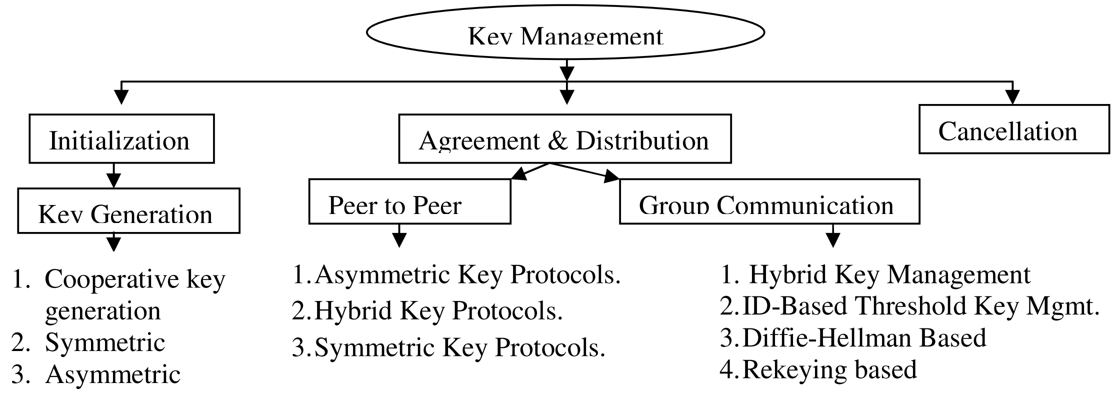

Figure 1: Key Management Protocol Taxonomy

3. KEY INITIALIZATION

First task of key management is how to generate the key, Manual or Automatic? In today's work manual generation of key is awkward and thus it should be automatic. First method of auto-key generation was by proposed by Arvid Damm in 1919 and that was used in World War II. Initialization of key in network security world depends on three major classifications:

cooperative key generation, symmetric key generation and asymmetric key generation. In cooperative key generation each member contributes to the key either through TTP or without TTP. The protocols falls in this category are group protocols based on: Elliptic Curve Cryptosystem, FireFly, RSA, Diffie-Hellman based protocols[10]. Another classification is based on symmetric or asymmetric initialization. The protocols in this category either lies on other protocol for key generation or generate key using discrete mathematics [11, 12]. We will study the mathematical details of key generation along with its agreement and distribution protocols.

4. KEY AGREEMENT & DISTRIBUTION PROTOCOLS

Key agreement is one or another form of key distribution [10]. The distinguishing feature is contribution from each member and the contributed part generates the key value. Key agreement can be used in place of key distribution, where users agree equally collects the contributed part to compute the shared key and this shared key is distributed to all members. Majority of key agreement protocols are based on Diffie-Hellman key agreement protocols and key distribution are either based on TTP or TTP less. In this section, various key agreement and distribution protocols are studied with execution details.

4.1 Peer to Peer Communication Based Key Management Protocols

4.1.1. Symmetric Key agreement & distribution protocols

If a party Alice wants to share a key with another party Bob then this can be achieved through number of ways using symmetric key distribution [13]. The possible methods are:

(a) **Method 1:** Alice computes some key and physically delivers it to Bob.
(b) **Method 2:** A third party can compute some key and physically distribute to Alice and Bob.
(c) **Method 3:** In a build secure session, either Alice or Bob encrypt new key with old key and deliver it to other party.
(d) **Method 4:** Alice and Bob each have a secure connection with third party and that third part securely distribute a key to Alice and Bob.

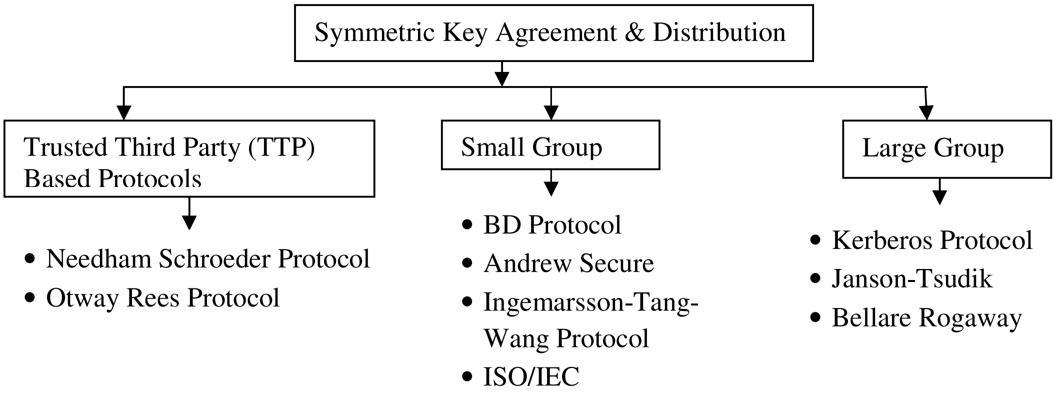
Symmetric Key Agreement & Distribution
Trusted Third Party (TTP) Based Protocols
Small Group
Large Group
• Needham Schroeder Protocol
• Otway Rees Protocol
• BD Protocol
• Andrew Secure
• Ingemarsson-Tang-Wang Protocol
• ISO/IEC
• Kerberos Protocol
• Janson-Tsudik
• Bellare Rogaway

Figure 2: Symmetric Key Agreement & Distribution Protocols

Method 1 and Method 2 have the advantage that data is exchanged only with the partner. Disadvantage of these methods are: (i) In a large distributed system, it is time consuming and difficult to manually transmit large number of keys. (ii) In a contributory network, manually collecting the shares increases the complexity of the overall system. (iii) In node level encryption or application/user/processes level encryption, millions of keys may be required. Method 3 has the advantage that (i) Limited physical transmission, (ii) Node level as well as application level encryption is possible. Disadvantages of this method are: (i) If an attacker is able to guess a key, he/she can easily evaluate the subsequent keys and (ii) Initial distribution of millions of keys persists. Method 4 has the advantages that (i) unique master key for each end system or user (ii) unique session key for each subsequent session (iii) time duration for session key can easily be manipulated in connection oriented protocols. The disadvantages of this method are: (i) Master key distribution is through physical delivery. (ii) In connectionless protocols, millions of keys need to be generated and generation should be rapid otherwise, it can easily be eavesdropped. Figure 2 show the classification of symmetric key agreement and distribution protocols.

4.1.1.1 TTP Based Protocols

Let Alice and Bob want to transmit data over a secure channel with session key K_S. K_A is the master key shared between Alice and Key Distribution Centre (KDC). Similarly, K_B is the master key shared between Bob and KDC. A nonce 'N' is used to represent the freshness of a key. Thus, it may be a timestamp, a counter, or a random number. Following are the steps to establish a secure channel.

Step 1: Alice (end-entity) makes a request to KDC (third party) for a session key.

Step 2: KDC distribute session keys to Alice and Bob as shown in Figure 3a or KDC distributes both keys to Alice and Alice pass the symmetric key to the target i.e. Bob as shown in Figure 3b.

Step 3: Secure communication channel is established using session keys.

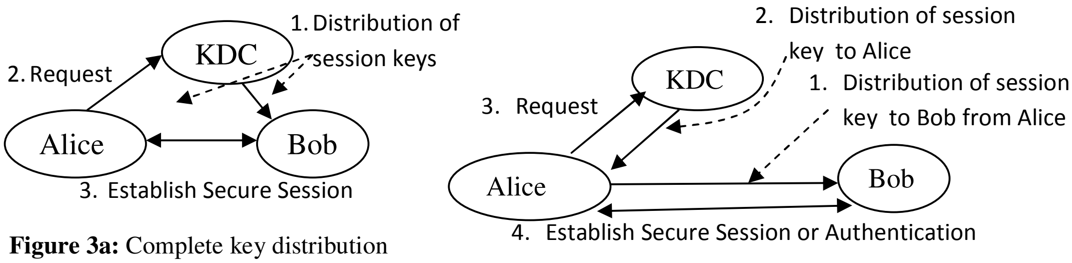
1. Distribution of
session keys
KDC
2. Request
Alice
Bob
3. Establish Secure Session
2. Distribution of session
key to Alice
KDC
1. Distribution of session
key to Bob from Alice
3. Request
Alice
Bob
4. Establish Secure Session or Authentication

Figure 3a: Complete key distribution through KDC.

Figure 3b: Key distribution through KDC and Party.

Figure3: Trusted Third Party Key Distribution methods

The advantage of TTP based protocol is that it prevents someone to eavesdrops the key and generate subsequent keys. Disadvantage of this protocol is initial bootstrapping of key distribution is mandatory [13]. The protocols based on TTP based protocol category are: (i) Needham-Schroeder Protocol [14, 15] (ii) Woo-lam Protocol (iii) Otway Rees Protocol.

4.1.1.2 Needham Schroeder Protocol

Roger M. Needham and Michael D. Schroeder proposed first technique for authentication and key distribution in 1978[15]. This protocol is based on the use of nonce and challenge/response handshake to verify freshness. Following are the steps of Needham Schroeder protocol:- (i) Alice→KDC: $E_{K_A}\{Alice, Bob, N_A\}$ to KDC. Where, N_A is nonce of Alice. (ii) KDC→Alice: $E_{K_A}\{N_A, Bob, K_S, E_{K_B}\{Alice, K_S\}\}$. Where, K_S will be the session key shared between Alice and Bob (iii) Alice→Bob: $E_{K_B}\{Alice, K_S\}$ (iv) Bob→Alice**:** $E_{K_S}\{N_B\}$ for authentication (v) Alice→Bob: $E_{K_S}\{f(N_B)\}$. Where, $f(N_B)$ may equals to N_B-1. This will indicate to Bob that this is not replay of step 3 message but an acknowledgement to step 4. Steps 1 to 3 are actually used for key distribution and steps 4, 5 and 3 are meant for authentication. D. Denning, G. Sacco in

1981 found the weakness of this protocol that Bob is not able to determine the freshness of key K_S. Any attacker for example, 'Trent' can impersonate Alice from a previous session. Trent can replay last three steps to provide Bob with the old key.

4.1.1.3 Otway Rees Protocol

This is another peer to peer symmetric key management protocol developed in 1987 [16]. This protocol prevents replay attack and eavesdropping. Let A and B are the two end parties, K_{ATTP}, K_{BTTP} are the shared key between end parties and trusted third party and K_S is the required symmetric key to be establish between end parties. The protocol runs as:

(i) A→B:$\{R, ID_A, ID_B, E_{K_{ATTP}}\{N_A, R, ID_A, ID_B\}\}$.

(ii) B→TTP: $\{R, ID_A, ID_B, E_{K_{ATTP}}\{N_A, R, ID_A,, ID_B\}, E_{K_{BT}}\{N_B\}, E_{K_{BT}}\{R,, ID_A,, ID_B\}\}$.

(iii) TTP→B: $\{R, E_{K_{ATTP}}\{N_A, K_S\}, E_{K_{BTTP}}\{N_B, K_S\}\}$. (iv) B→A: $\{R, E_{K_{AT}}\{N_A, K_S\}\}$. Where, R is run identifier between A and B.

Weakness of this protocol is impersonation attack found in Boyd and Mao [17, 18].

4.1.1.4 Other Protocols

Some other protocols in this category are [19]: (i) Wide-Mouthed-Frog (WMF) Protocol: This protocol was proposed in 1989 by Burrows et. al. The protocol runs as: (i) A→TTP: $ID_A, E_{K_{ATTP}}\{T_A, ID_B, K_S\}$ (ii) TTP→B: $E_{K_{BT}}\{T_{TTP}, ID_B, K_S\}$. Major strengths of this protocol are: freshness and identity check. Weaknesses of this protocol are: (a) replay attack, (b) no efficient forward or backward secrecy, (c) no mutual authentication.

(ii) Kerberos Protocol: Versions of this protocol varies from v1 to v6. It uses the concept of timestamp instead of challenge-response but it is based on Needham Schroeder protocol. Strengths of this protocol are: (a) provide authentication as well as authorization, (b) no replay attack, (c) more protected against forward and backward secrecy. Weaknesses of this protocol are: (a) required multiple TTP to control the system. (b) proper synchronization of time clock.

(iii) Janson-Tsudik Three Party Protocol: This protocol was proposed by Janson and Tsudik in 1993. In this protocol, Alice-TTP and TTP-Bob establish a two party key distribution protocol. Handshaking between Alice and Bob is done there after for authentication. Strengths of this protocol are: (a) mutual authentication and confirmation. (b) Proper freshness checking but this protocol does not provide forward secrecy.

(iv) Bellare-Rogaway Three Party Protocol: This protocol is proposed by Bellare and Rogaway in 1995. It checks key freshness and authentication. It is also efficient in terms of computations and number of rounds. This protocol does not provide entity authentication or key confirmation.

Some other protocols without TTP are compared in table 1.

Table 1: Protocols with support of TTP [19].

Protocol	Year	Strengths	Weaknesses
Andrew Secure RPC	1989	1. Handshaking with use of nonce. 2. Old key is used to encrypt new session key.	1.No backward secrecy. 2.No assurance about freshness of key.
Janson-Tsudik (Two-Party)	1993	1. Minimum rounds and operations. 2. Both Authentication and confidentiality is checked and protected against forward secrecy 3. Two-party key distribution protocol.	1. Weak against forward and backward secrecy.

ISO/IEC 11770-2		1. Seven TTP based and six with TTP based protocols. 2. Protocols are encryption, timestamp and nonce based simple ptotocols. 3. Old key is used to generate new key	1. Very basic protocols. 2. No proper authentication checking. 3. Forward or backward secrecy is protected.

4.1.2 Hybrid Key Management Protocols

4.1.2.1 Beller-Chang-Yacobi (BCY) & Beller-Yacobi (BY) Protocol

Beller’s original protocol is based on hybrid cryptosystem using symmetric and asymmetric cryptographic algorithms. The Rabin public key cryptography [20] based on modulo square root (MSR) algorithm is computationally mobile communication efficient technique and this technique is having three variants. The variants are discussed below:

(i) Basic version of MSR algorithm uses private key KR_i , public key KP_i , secret certificate SC_i and session key SK_i of entity ‘i’. Following are the steps of basic MSR algorithm: 1. Base ‘B’ send its identity and public key PK_B to Mobile ‘M’ $(B \rightarrow M: ID_B, PK_B)$ 2. M generate session key SK_{BM} , encrypt using public key and send it to ‘M’ $(M \rightarrow B: E_{PK_B}\{SK_{BM}\})$ 3. M sends its identity ID_M and secret certificate SC_M , encrypted using symmetric key to B. Carlsen[21] identified various weakness in this protocol and the weaknesses are: (a) masquerading of ID_B and (b) no mechanism to protect against reuse of old key by an attacker.

(ii) Second version of the protocol is named as improved MSR (IMSR). Beller [22] and Carlsen[21] independently proposed IMSR. IMSR is having following steps: 1. $B \rightarrow M: ID_B, N_B, PK_B, Cert(B)$. Where N_B is nonce, Cert(B) is certificate of B. 2. $M \rightarrow B: E_{PK_B}\{SK_{BM}\}$ 3. $\rightarrow B: E_{SK_{BM}}\{N_B$, M, $SC_M\}$. Weaknesses of this mechanism are (a) impersonation attacks pointed out by V. Vardharajan and Y. Mu [23, 24]. (b) More complex than basic MSR. Another modification made by C. Boyd and A. Mathuria[25] is: 1. $B \rightarrow M: ID_B, N_B, PK_B, Cert(B)$. 2. $M \rightarrow B: E_{PK_B}\{SK_{BM}, N_B, SC_M\}$ 3. $M \rightarrow B: E_{SK_{BM}}\{ID_M\}$.

(iii) Third version of the protocol integrates Diffie-Hellman protocol [25] with MSR. Here, symmetric and session keys are generated with the help of primitive elements. Steps of MSR+DH protocol are: 1. $B \rightarrow M: \{ID_B, N_B, PK_B, Cert(B)\}$. 2. $M \rightarrow B: E_{PK_B}\{SK_{BM}\}, E_{SK_{BM}}\{N_B, ID_M, PK_{M,}Cert(B)\}$. Weakness of this protocol is that it is more complex and requires hundred time more computation than any other version of the protocol.

4.1.2.2 Beller-Yacobi’s (BY) Protocol

In another work, Beller and Yacobi proposed another improvement over MSR using offline pre-computation of ElGamal digital signature [26] and challenge response mechanisms. It shows changes in last two steps and the protocol run as: 1. $B \rightarrow M: ID_B, PK_B, Cert(B)$ 2. $M \rightarrow B: E_{PK_B}\{SK_{BM}\}$ 3. $B \rightarrow M: E_{SK_{BM}}\{N_B\}$ 4. $M \rightarrow B: E_{SK_{BM}}\{ID_M, PK_M, Cert(M), DS_{PK_M}\{N_B\}\}$. Where, $DS_k(x)$ represents digital signature of x using key ‘k’. Weakness of this protocol is that a friend of either B or M can start a parallel session attack [25]. This attack can be protected using fresh digital signature and hashing mechanism.

4.1.2.3 Tatebayashi, Matsuzaki and Newman (TMN) Protocol

TMN is computationally efficient, mobile to mobile based hybrid key management protocol [27]. The protocol takes help of distributed server (DS) to share a session key. The protocol run as follows: 1. $M \rightarrow DS{:}\, ID_B, E_{PK_S}\{SK_M\}$ 2. $DS \rightarrow B{:}\, ID_A$ 3. $B \rightarrow DS{:}\, ID_A, E_{PK_S}\{SK_B\}$ 4. $DS \rightarrow M{:}\, ID_B, E_{SK_M}\{SK_B\}$. Weaknesses of this protocol are: parallel session attack [28], replay attack [28] and secrecy attack [29, 30]. Fourth step was later modified and common session key was considered as contribution from SK_M as well as SK_B. For example: SK_M XOR SK_B.

4.1.2.4 Yacobi and Shmuely (YS) Protocol

Yacobi and Shmuely designed a hybrid key management protocol for wireless communication in 1989[31]. This protocol provides link security only. Authentication of ends is not escrowed on any party system. If SK_B and SK_M are the session key parts selected by base and mobile and r_B, r_M are the random number selected by base and mobile respectively then protocol runs as: (i) B→M: SK_B+r_B (ii) M→B: SK_M + r_M. Session key SK_{BM} = $(g^{x_B+r_B} y_B)^{r_M}$= $(g^{x_M+r_M} y_M)^{r_B}$= $g^{r_M r_B}$. Weaknesses of this protocol are: Martin and Mitchell's old key reuse attack, Boyd and Park's no signature effect attack and masquerading attack of mobile or base in order to communicate with other members.

4.1.2.5 Park's Protocol (PP)

C. S. Park made modification on YS protocol in 1997[32]. The protocol run as: (i) B→M: $\propto^{SK_B+r_B}$ (ii) M→B:$SK_A + r_A$ where, α is a primitive element. Weaknesses of this protocol is Martin and Mitchell's old key use attack, masquerading attack but only on base side and Boyd & Park's no signature effect.

4.1.3 Public / Asymmetric Public key Management Protocol

The public key cryptography/asymmetric key cryptography has been evolved from an assumption made by James H. Ellis in 1970 that the knowledge of an encryption key may not necessarily lead to an efficient derivation of the decryption key. The practical feasibility of this idea was proven by multiplication of large prime numbers which lead to non-invertible results in 1973 by Clifford Cocks. This concept was first published by W. Diffie and M. Hellman in 1976, which later became popular as Diffie-Hellman key agreement protocol.

In public /asymmetric key cryptosystem, keys can be distributed either through some centralised or decentralised mechanism. In centralised mechanism, first approach is through a central directory. Like a telephone directory, the central directory stores public keys of each user. If any user wants to communicate with another user, he/she can obtain public key of the destination user and encrypt the data using that public key. Disadvantage of this method are: (i) heavy traffic towards the central directory, (ii) no perfect backward or forward secrecy, (iii) impersonating the user and eavesdropping the message is comparatively easy. Second centralised mechanism is through public authority. Here, directory is replaced with authority. Public authority will also have private/public key. On request from a user, the authority encrypt the public key of user with the private key of the authority. Similarly, all users can obtain public keys. It is seven steps communication, in which, last two steps are to check the freshness of the keys. This method is having its own disadvantages like: (i) any user can pretend to be authority in order to eavesdrop the information, (ii) no strong protection against forward or backward secrecy and (iii) centralised asymmetric key cryptography is based on key certificates. A certificate authority issue certificates to the authenticated users those having public key of authority. This method freshness is checked through a proper timestamp. The disadvantage of this method is that any user can replay old certificates to obtain the updated information. Various protocols listed in this category are described below.

4.1.3.1 Aziz-Diffie (AD) Protocol

AD protocol is heavy public key cryptography based mechanism [25, 33]. In this protocol, M and B has public key certificates signed from trusted certificate authority. Like other protocols, session key SK_{BM} is generated from shares of B and M, AL_M is list of shared key algorithms supported by Mobile; SL_B is shared key algorithm selected by entity B. The protocol is having following steps: 1.$M \rightarrow B: Cert(M), N_M, AL_M$. 2. Hashing function H (x) is used over digital sign and $B \rightarrow M$: $Cert(B), E_{PK_M}\{SK_B\}, SL_B, E_{PR_B}\{H(E_{PK_B}\{SK_M\}, SL_B, N_M, AL_M)\}$ 3. $M \rightarrow B: E_{PK_B}\{SK_M\}, \{H(E_{PK_B}\{SK_M\}, E_{PK_M}\{SK_B\})\}$. Weakness of this protocol is Meadow's attack [34].

4.1.3.2 Advanced Security for Personal Communications Technologies (ASPeCT) Protocol

This is an asymmetric key cryptography based authentic and key establishment protocol that is particularly used in UMTS (Universal Mobile Telecommunications System). Mobile 'M' and Base 'B' take help of TTP [35, 36, 37]. Let α is the primitive element and $\propto^{r_M}, \propto^{r_B}, \propto^{r_{ttp}}$ are the public key agreement of mobile, base and TTP, where r_i is the random number selection by the entity 'i'. Functions h1, h2 and h3 are one way hash functions. C(B), C(M) are the certificates of base and mobile respectively. CC(X,Y) denotes the sequence of certificates i.e. C_0, C_1, ... C_n. These sequence of certificates are all digitally signed through a proper mechanism [35], T_S is the time stamp, cidI is the certificate identity of entity I. Ch_data is the tariff information and pay_data is the payment data. The protocol runs as: (i) M→B:$\propto^{r_M}, ID_{TTP}, E_{K_{MTTP}}(ID_M)$. (ii) B→TTP: $\{\propto^{r_M}, E_{K_{MTTP}}(ID_M), C(B)\}$. (iii) TTP→B: T_S, CC(M,B), $E_{K_{MTTP}}(CC(B,M))$, CC(B,TTP), $E_{K_{MTTP}}\{Sig_{TTP}(h3(\propto^{r_M}, cidM, cidB, T_S))\}$. (iv) B→M:$\{r_B, h2\ (K_{MB}, r_B, ID_B), ch_{data}, T_S, CC(M,B), E_{K_{MTTP}}\{Sig_{TTP}(h3(\propto^{r_M}, cidM, cidB, T_S))\}$. (v) M→B: $E_{K_{MB}}\{Sig_M(h3(\propto^M, \propto^B, r_B, ID_B, ch_{data}, T_S, \propto_T, IV)), \propto_T, IV\}$. Where, IV is the random number initial vector and $\alpha_T = F_{IV}^{T}(\propto_0)$ is the initial value for the tick payment.

4.1.3.3 Other Protocols

Some other protocols in public key cryptography are compared in Table 2.

Table 2: Pubic/ Asymmetric key management protocols.

Protocol	Year	Strengths	Weaknesses
MTI [38, 39]	1986	• Based on Diffie Hellman key agreement. • Two special cases of three infinite families of key agreement protocol. i.e. MTI/A0 and MTI/C0. • Communication is faster because of least computation • Complexity of MTI/C0 is 3M+ 1I and MTI/A0 is 3M +1A. Where M is multiplication, I is inversion and A is addition	• MTI/A0 and MTI/C0 are vulnerable to small subgroup and unknown key share attack. • Forward secrecy is weak
MQV [40, 39]	1998	• Protected against small subgroup and known key share attack. • Elliptic curve based cryptosystem. • Medium forward secrecy, faster communication. • Certificate based hybrid key management.	• Complexity is 3M+1A. • Unknown key share attack [41].

Scott [42, 39]	2002	• ID based key management. • Secure against impersonation. • Complexity is 9 operations per single round.	• Parallel session attack. • Not protected against masquerading.
Smart [43, 39]	2002	• ID based key agreement. • Fast Communication. • Complexity is 8 operations per single round.	• Partial forward secrecy, impersonation, unknown key share resilience, known key security.
Chen and Kundla [44,39]	2002	• ID based key management. • Complexity is 9 operations per single round.	• Partial forward secrecy, unknown key share resilience and key impersonation.
Shim [45, 39]	2003	• ID based key management. • Complexity is 10 operations per single round.	• Man in middle and parallel session attack.
McCullagh and Barreto [46, 39]	2004	• ID based authenticated key agreement protocol. • Modes: Escrow or Escrowless. • 7 operations per single round. • High forward secrecy, no impersonation, known key security.	• Parallel Session attack.
Jeong, Katz and Lee [47, 39]	2004	• Diffie Hellman based three key agreement schemes i.e. TS1, TS2, TS3. • TS1 is not forward secrecy protected but TS2 and TS3 are protected. • It is based on dynamic, centralized diffie hellman approach. • Fast communication and medium key size. • Certificate based approach. • Complexity is 3E+1MC, where E is exponentiation and MC is MAC calculation.	• No time stamp and thus prone to parallel session attack.

4.2 Group Communication Based Key Management Protocol

Another classification of key agreement protocols is based upon Group Communication (GC) [48, 49]. In GC, message is sent by a single valid group member and multiple valid members can receive the message. GC ensures that Group Key Management (GKM) provides key access to valid group members only [49]. Further, GKM makes sure that (i) it is infeasible for any malice user to computationally calculate group key and (ii) if any user is known of subset of keys, he/she can evaluate subsequent or preceding group keys. GKM involves key agreement and key distribution. The protocols used for key management are: (i) Group Diffie Hellman (GDH) (ii) Centralized Key Distribution (CKD) (iii) Tree-Based Group Diffie-Hellman (TGDH) (iv) Skinny Tree (STR), (v) Burmester-Desmedt (BD), (vi) Octopus.etc.

4.2.1. Diffie-Hellman Based Protocols

The idea of System I called ax1x2 was developed by Whitfield Diffie and Martin Hellman [50]. Now this system is known as Diffie-Hellman key agreement protocol [50]. It was proposed by Whitfield Diffie and Martin Hellman in 1976[51]. Later on there were many attempts to extend this protocol in different directions. Pre-computation in generic protocol assumes that all, 2-parties to n-parties, agree on a cyclic group 'G', of order 'q' and a primitive element 'α' of this group G. Choose a large prime number 'p' such that 'p' and 'α' are publically known to all.

4.2.1.1. 2-Party Diffie-Hellman

According to Michael Steiner, Gene Tsudik and Michael Waidner [52], in 2-party case, M^i and M^{3-i} where i= {1, 2}, both parties randomly select integers y^i such that private key $K_{PR}{}^i$= y^i and public key $K_{PUB}{}^i$=α^{y^i}mod p. Both parties compute shared key K= $\alpha^{y^i y^{3-i}}$mod p. In n-party case, M^1….. M^n, protocols related to Diffie-Hellman Key Agreement in Group Key Distribution (GKD) are: GDH.1, GDH.2 and GDH.3 [52].

4.2.1.2. N-Party Group Diffie-Hellman(GDH) (N>2)

According to Steiner, Tsudik and Waidner, GDH.1 consists of two stages: (i) Upflow and (ii) Downflow. In upflow stage M^i computes the set $\{\alpha^{\pi(y^k|k\epsilon[1,j])}|j\epsilon[1,i]\}$ and send it to M^{i+1}. When the last party receives the upflow message then it computes group key K_n = $\{\alpha^{y^1 \ldots\ldots y^{n-1}}\}^{y^n}$. In downflow stage M^{n-i+1}computes the set $\{\alpha^{\pi(y^k|k\notin[i,j])}|j\epsilon[1,i]\}$ and send it to M^{n-i}having (n-i) values. Major drawbacks of GDH.1 are its large number of rounds and no special communication requirements like: broadcast or synchronization [52].

GDH.2 also consists of two stages: (i) Upflow and (ii) Broadcast. In upflow stage, M^1 compute the set $\{\alpha^{y^1}\}$ and send it to M^2. M^2 compute the set $\{\alpha^{y^1}, \alpha^{y^2}, \alpha^{y^1 y^2}\}$ and send it to M^3. M^3 compute the set $\{\alpha^{y^1 y^2}, \alpha^{y^1 y^3}, \alpha^{y^2 y^3}, \alpha^{y^1 y^2 y^3}\}$ to M^4 and so on. If it is n=4 party system then M^4 will be the last party. M^4 computes the set $\{\alpha^{y^1 y^2 y^4}, \alpha^{y^1 y^3 y^4}, \alpha^{y^2 y^3 y^4}\}$ and in broadcast stage it broadcast this set to all group members. Finally, group members compute the shared group key S_4= $\{\alpha^{y^1 y^2 y^3 y^4}$mod p}. The major advantages of GDH.2 are minimum number of rounds as compared to GDH.1 i.e. almost half, minimum number of messages sent / received by each participant, small number of exponentiation, secure and easy to implement [52].

GDH.3 consists of four stages: (i) Upflow (ii) Broadcast (iii) Response and (iv) Broadcast. In upflow stage, M^1 computes the set $\{\alpha^{y^1}\}$ and send it to M^2. M^2 compute the set $\{\alpha^{y^1 y^2}\}$ and send it to M^3. M^3 compute the set $\{\alpha^{y^1 y^2 y^3}\}$ and send it to M^4. If M^4 is the last party, then it sets shared group key S_4= $\{\alpha^{y^1 y^2 y^3 y^4} \; mod \; p\}$. In second stage, M^4 broadcasts $\{\alpha^{y^1 y^2 y^3}\}$ to everyone. In third stage, parties compute inverse of their shares and reply back to last new node. For example, M^1, M^2, M^3 compute inverse and send $\{\alpha^{y^1 y^2}\}$, $\{\alpha^{y^1 y^3}\}$, $\{\alpha^{y^2 y^3}\}$ to M^4 respectively. In the last stage, M^4 broadcasts $\{\alpha^{y^1 y^2 y^4}, \alpha^{y^1 y^3 y^4}, \alpha^{y^2 y^3 y^4}\}$ to everyone. Major advantages of GDH.3 are: constant message size and constant with less number of exponentiation for M^i as compared to GDH.1/2.

4.2.1.3. Authenticated-GDH (A-GDH)

Gluseppe Ateniese, Michael, and Gene Tsudik extend the GDH to Authenticated-GDH[53]. In Authenticated 2-party Key Agreement case, M^i and M^{3-i}, where i= {1, 2}, M^i party randomly selects integers y^i such that private key $K_{PR}{}^i$= y^i and public key $K_{PUB}{}^i$=α^{y^i}mod p. M^i sends $K_{PUB}{}^i$

to M^{3-i}. M^{3-i} sends $K_{i(3-i)}$=$\alpha^{K_{(3-i)i}y^{(3-i)}}$mod p to M^i. Where, $K_{(3-i)i} = F\left(\alpha^{y^i y^{(3-i)}} mod\, p\right)$. Where $F(x) = h(x)$ and h is a hash function, $h{:}\,\{0,1\}^* \rightarrow Z^*{}_q$.F(x) The public values of the system are (p, q, α, α^{y^1}, α^{y^2}). After M^i calculates inverse with $K_{i(3-i)}$ the shared key for M^i and M^{3-i} becomes S_2= $\alpha^{y^i y^{(3-i)}} mod\, p$. This protocol is extended to Authenticated n-party GDH (A-GDH). In A-GDH, GDH.2 is extended because of its intractability of key agreement and in terms of minimum of total number of messages [53]. Although it is assumed that these changes can easily be adapted to GDH.3. According to A-GDH.2, first stage is similar to GDH.2 and in second stage, authentication is provided using key inverse operation (K^{-1}). In second stage i.e. broadcast, last party i.e. M^n broadcast $\{\alpha^{(y^1 \ldots\ldots y^n)(y^i)^{-1}K_{in}}\ \}$ to M^i. where K_{in}=F($\alpha^{y^i y^n} mod\, p$). At M^i S_n is computed as $\{\alpha^{(y^1 \ldots\ldots y^n)(y^i)^{-1}K_{in} y^i K_{in}^{-1}}$ mod p} = $\{\alpha^{(y^1 \ldots\ldots y^n)}$mod p}. This method provides perfect forward secrecy but all authentications are performed with M^n and this M^n will not make sure of other members about confidentiality of arbitrary nodes.

4.2.1.4. Complete GDH (SA-GDH)

To make sure that each member is aware of membership, complete group key agreement (SA-GDH.2) is presented. During upflow stage in this protocol, it broadcast $\{\alpha^{y^1 y^2}, \alpha^{y^1 y^3}, \alpha^{y^1 y^3}, \ldots\ldots \alpha^{y^1 y^n}\}$ to next node. Second node completely authenticates this using $\{\alpha^{y^i y^j}, \alpha^{y^j y^i}\}$ transmission to every next node. This type of communication is also based on version of GDH. Strengths of SA-GDH are perfect forward secrecy, mutual authentication, protected against old key reuse attack. Weaknesses of this protocol are: large number of exponentiation operations and keys generation per unit time.

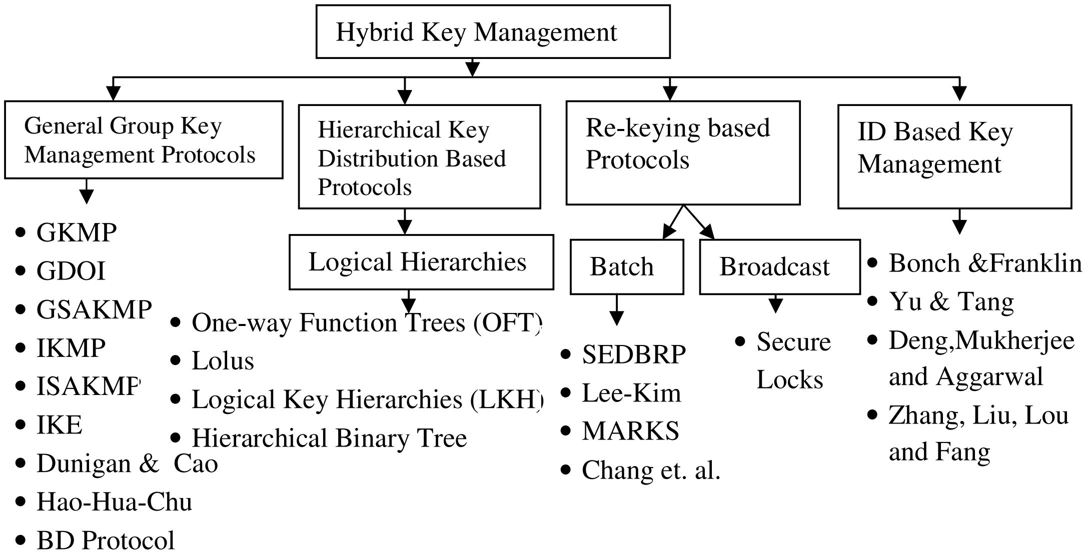
Hybrid Key Management
General Group Key Management Protocols
Hierarchical Key Distribution Based Protocols
Re-keying based Protocols
ID Based Key Management
• GKMP
• GDOI
• GSAKMP
• IKMP
• ISAKMP
• IKE
• Dunigan & Cao
• Hao-Hua-Chu
• BD Protocol
Logical Hierarchies
• One-way Function Trees (OFT)
• Lolus
• Logical Key Hierarchies (LKH)
• Hierarchical Binary Tree
Batch
Broadcast
• SEDBRP
• Lee-Kim
• MARKS
• Chang et. al.
• Secure Locks
• Bonch &Franklin
• Yu & Tang
• Deng,Mukherjee and Aggarwal
• Zhang, Liu, Lou and Fang

Figure 4: Classification of Group Key Management Protocols

4.2.2. Hybrid Key Agreement and Distribution Protocols

In hybrid technique, public/asymmetric key cryptosystem is used to transmit secret/private/symmetric key. In a simple hybrid form, two users communicate among themselves to exchange keys. On request, secret key is encrypted using requester's public key. The major disadvantage of this technique is freshness. No nonce or timestamp mechanism is integrated. Enhancement to this mechanism is made through the use of nonce or timestamp. This added confidentiality and authentication. Another hybrid approach is with the use of KDC. KDC distributes secret key to every user. Using this secret key, session keys are encrypted/decrypted. Public key cryptography is integrated with KDC to distribute secret key. This approach is used on IBM mainframes. Figure 4 shows the classification of hybrid key agreement and distribution protocol. Due to space constraint, some protocols are discussed in this section and others are out of scope of this paper.

4.2.2.1. General Group Key Management Protocols

4.2.2.1.1 Group Key Management Protocol (GKMP)

Various network security services provided by GKMP are access control, key generation and key distribution [54, 55]. GKMP is a decentralized key distribution protocol. This protocol creates group symmetric keys but there is no need of centralized KDCs. GKMP has various entities to manage the network. The entities are: (i) Group Controller (GC), (ii) Group Member (GM), (iii) Group Key Packet (GKP), (iv) Group Traffic Encryption Key (GTEK), (v) Group Key Encryption Key (GKEK), (vi) Group Re-key Package (GRP), (vii) Session Key Package (SKP), (viii)Compromised Recovery List (CRL) and (ix) Session Key Encryption Key (SKEK) [55]. This protocol works in thirteen states and these states are divided among following four functionalities: (i) Group Key Creation, (ii) Group Re-key, (iii) Member initiated Join and (iv) Member Deletion [55]. Table 3 shows the working of GKMP. First group member makes a command to GC to establish a group and in second function group key creation start using Create_group keys_1 from GC to GM. This function is having four communications. Data Primitives used are the parameters set inside a packet. These primitive elements are included in the subsequent communication packets also. After key generation, key distribution process starts. This function is having six communications to distribute the keys among GMs. In Re-keying function, if some key loses some message due to disruption, information warfare then, to maintain the data integrity re-keying process start. This process is same as group key creation. Here, GM verifies the keys and authenticates other keys also. Finally, GC is to multicast the updated keys. If a new member wants to join the group then member joins initiation start. This function is having seven communications. GM makes a request to GC and GC establishes a SKEK to support the secure group establishment and communication. Remaining five communications are same as group key distribution. Finally member deletion can either be hostile or cooperative. The cooperative member deletion is more reliable than hostile. Cooperative member deletion occurs among a trusted GM and GC whereas hostile member deletion occurs if some GM losses it's trust in the group.

Table 3: Group Key Management Protocol

Function	**Source State**	**Dest. State**	**Message definition from source to destination**	**Data Primitive Used**
Initiation	Any node		Create Group command to GC	Group ID (GID), GCID
Group Key Creation	State 1 (GC)	State2 (GM)	Create_group keys_1	GID, GTEK ID, GKEK ID, GTEK Creation, GKEK Creation, GC public key
	State 2 (GM)	State 2 (GC)	Create_group keys_2	GID,GTEK ID, GKEK ID, GTEK Creation, GKEK Creation, GM public key
	State 2 (GC)	State 3 (GM)	Negotiate Group Keys_1	GID, GTEK ID, GKEK ID, GC Access Control
	State 3 (GM)	State 4 (GC)	Negotiate Group Keys_2	GID, GTEK ID, GKEK ID, GM Access

				Control
Group Key Distribution	State 4 (GC)	State 5 (GMs)	Create Session KEK_1	GID, GTEK ID, GKEK ID, GCID , GM Access Control, GC public key
	State 5 (GM)	State 5 (GC)	Create Session KEK_2	Random value, other member public key
	State 5 (GC)	State 6 (GM)	Negotiate Session Keys_1	GID, SKEK ID, CRL version number, GC Access Control, group token
	State 6 (GM)	State 7 (GC)	Negotiate Session Keys_2	GID, SKEK ID, CRL version number, GC Access Control, group token
	State 7 (GC)	State 8 (GMs)	Download Group Keys	GID. SKEK ID
	State 8 (GMs)	State 9 (GC)	Key Download ACK	
Group Re-Key	Re-keying is a two step function. In first step i.e. communication between GC and GMs is same as Group Key Creation. Second step i.e. Rekey Multicast is described in next step			
	State 4 (GC)	State 10 (GMs)	Rekey Multicast	GID, GTEK ID, GKEK ID, GC Access Control
Member Initiated Join	GM	State 11 (GC)	Request_Group_Join	Request, GID, GM public key
	State 11 (GC)	State 5 (GM)	Create Session KEK_1	GID, GTEK ID, GKEK ID, GCID , GM Access Control, GC public key
	Rest of the communication is same as in group key distribution			
Hostile Member Deletion	GC	State 12	Delete_Group_Keys	GID, Request, GC public key
	State 12 (GM)	State 9 (GC)	Group_Keys_Deleted_ACK	GID, GM ID, GM public key
Cooperative Member Deletion	GC	State 12	Delete_Group_Keys	GID, Request, GC public key

The advantages of GKMP are: (i) it provides flexibility over KDC to create multicast groups in which single transmission can be received by multiple GMs. (ii) any member in the network can create its own group and make secure multicast communication. (iii) Latency as compared to KDC group creation process is much less. Since in KDCs there is need to take permission prior to establish a group. Whereas, in GKMP each member can create it's own autonomous group. (iv) Extendibility of network groups is cumbersome in TTP mechanisms. Whereas in GKMP there is no overhead on any central party and thus it can process its own permission control based on certificates. (v) TTP is a centralized mechanism and thus demands high security procurements for the centralized server. This increases the cost of security. But in GKMP security of keys is to the members only and thus does not demand any extra cost. (vi) In TTP, a large number of requests come to centralized resource and create a bottleneck, Whereas, GKMP does not have any bottleneck. (vii) Failure of centralized resource in TTP makes the system down but GKMP's distributed nature helps in recreation of groups. (viii) Security of GKMP is high as compared to TTP because of small sized groups and lesser involvement of large number of un-trusted members. The disadvantages of GKMP are: (i) groups in this protocol are not scalable because a large size group further creates a centralized scenario. (ii) It does not provide any mechanism to protect against perfect forward secrecy and subsequent keys can easily be generated. (iii) As this is a distributed approach, trusts among third party and GM is less.

4.2.2.1.2 Group Secure Association Key Management Protocol (GSAKMP)

GSAKMP is Diffie-Hellman key exchange based key management protocol [56]. The recommendations on Diffie-Hellman key exchange protocol is also taken into considerations while policy establishment and security parameters negotiation during implementation. The extensive architecture does not dilute the strength of the protocol because of Diffie-Hellman parameters but it adds more security suites to enhance the security services. It also inherits features from ISAKMP and FIPS Pub 196 for group communication and authentication. If some GMs are found to be compromised then GSAKMP may use logical key hierarchy (LKH) to destroy the older key and rebuild a new one. The goal of this protocol is to establish a trust among group members to equally protect the data among all communication. In order to achieve secure communication the entities used in this protocol are: (i) Group Member (GM), (ii) Group Owner (GO), (iii) Group Controller Key Server (GC/KS) and (iv) Subordinate GC/KS (S-GC/KS). The GO entity is the policy creation authority and develop policy token (PT) with an owner signature on it. The GC/KS performs creation of keys, distribution of these keys; rekey mechanism and group membership management using LKH. Subordinate GC/KS (S-GC/KS) performs the same functionality as GC/KS except creation of GTPK. It assumes that GTPK will come from GC/KS. Also, being as member with GC/KS, an additional task of verifying the authority of the GC/KS has to be performed. GM is an important entity in GSAKMP. GM can be classified as: (i) Normal GM (NGM) and (ii) Speaker GM (SGM). GM task is to properly check all the security related actions and use of group keys. It also checks that the GC/KS and S-GC/KS are authorized. GSAKMP offer mechanisms to select either one SGM or multiple SGMs. In one SGM case, all GMs will share a single GTPK and other security associated state information with SGM. Whereas in multiple SGMs, all SGMs share a common TPK and GMs can transmit GTPK to any SGM.

GSAKMP follows a life cycle to manage groups. The four phases of life cycle are: (i) group definition, (ii) group establishment and (iii) security relevant group maintenance. Group definition is a pre computation step to establish a secure group. This involves specifying the security parameters in a policy token. The major parameters in policy token are: (a) Token ID, (b) Authorization, (c) Access Control, (d) Mechanisms and (e) Signature. Token ID contains version number, protocol used, group identification and timestamp details. Authorization contains the group owner, key server and compromise recovery agent. Access control contains the user right and permission details. Mechanism specifies that whether it is a group communication or unicast security association. Signature block stores information about digital signatures and certification information. Group establishment is the process consists of three mandatory and two optional implementation steps: (i) Request to join, (ii) Key Download and (iii) Key Download Ack/Failure, (iv) Request to join error (optional) and (v) Lack of ACK messages. Like GKMP, communication in this protocol is occurring in between GM and GC and the steps are: (i) GM→GC: Request to join (contain group identification, nonce, key creation, signature of group member as mandatory and certificate, cookies, mechanism as optional parameters. (ii) GC→GM: Key Download (contain identification, key creation, encrypted policy token, vendor identification, signature of GCKS, certificates etc.). (iii) GC→ GM: Request to join error (in case of unaccepted request to join message and it contain group identification, nonce, vendor identification etc.). (iv) GC→GM: Key Download ACK/Failure (contains group and vendor identifications, nonce, signature, acknowledgements etc.). (v) GC→GM: Lack of ACK message (group and member identification, nonces, signature of

GCKS, fields used in signature etc.) Weaknesses of this protocol are: (a) weak selection of randomly chosen parameters can deteriorate its performance. (b) System clock should not skew across a maximum limit. (c) No identity protection (d) not protected against replay attack. Strengths of this protocol are: (a) protection against denial of service attack. (b) strong time dependent key generation and (c) no tolerance against compromise of S-GC/KS or GO.

4.2.2.1.3 Group Data of Interpretation (GDOI) Protocol

GDOI is a two protocol based group security association Diffie-Hellman key management protocol [57, 58]. The two protocols are: (a) GROUPKEY-PULL and (b) GROUPKEY-PUSH. GROUPKEY-PULL protocol is used by GM to request policy information and keying material. GROUPKEY-PUSH protocol is used by GCKS to distribute this material. GDOI uses the support of Internet Security Association and Key Management Protocol (ISAKMP)[7] for key establishment and Internet Key Exchange (IKE) for secure channel[6, 59, 60]. GDOI uses IKE’s phase 1 and redefine phase2 with protocol GROUPKEY-PULL. In phase1 two peers establish a channel similar to ISAKMP. During new phase2 communication, four exchanges are performed for policy information and keying material. For updating SA, it uses protocol2 i.e. GROUPKEY-PUSH. The functionality of this protocol is divided among two operations: (a) authorization and (b) announcement. Authorization is meant for group policy and announcement is to listen requests from secure groups or members.

4.2.2.1.4 Dunigan and Cao (DC) Protocol

Dunigan and Cao [61] proposed a decentralised hybrid key management protocol. It generate token similar to GKMP but independently for subgroups. As compare to GKMP, this protocol claims to have features like key escrow, portable for small groups, better ticket management and secure Association / Security Parameter Index (SA/SPI) based protocol. Dunigan and Cao protocol are having same functional steps as of GKMP. Weakness of this protocol is that it does not justify how to deploy the proposed scheme over a large group network.

4.2.2.1.5 Hao-Hua-Chu (HHC) Protocol

In this protocol special intension is drawn to achieve following goals: (i) independence of source and destination from intermediate resources, (ii) portable to any multicast protocol (iii) efficient and graceful dealing with packet loss and delay channels, (iv) watermarking techniques to integrate source and destination identifications. It also helps in identifying the leaker’s information, (v) prevention against parallel session attack. And (vi) computationally efficient. The protocol runs as follows: 1. Sender generates two different watermarks of video frames. 2. A group leader generates traffic encryption key (TEK) using this key group leader can only decrypt random key ‘k’ selected by the source. 3. Group leader decrypts key ‘k’ and encrypts using receiver’s public key. 4. Digital verification is performed at receiver as well as group leader side for proper authentication. Weakness of this protocol is the large number of packets generation during rekeying process.

4.2.2.1.6 BD Protocol

Burmester Desmedt Group Key Agreement (BD GKA) protocol was proposed by M. Burmester and Y. Desmedt in 1994[62, 63]. Strength of this protocol is: protection against strong forward secrecy. Weaknesses of this protocol are: inefficiency of key computation, large number of communication messages and signature verification messages as compared to Dutta and Barun‘s group key agreement protocol (DB GKA)[64].

4.2.2.2 Re-Keying Based Protocols

4.2.2.2.1 Secure Locks

Chiou and Chen proposed hybrid key cryptography based protocol [65]. Here, secure lock feature is implemented using Chinese Remainder Theorem (CRT). The protocol runs as follows:

1. Randomly select session key for encryption (SK^E) and decryption (SK^D) for a secret message 'M'. 2. Using CRT and public key of the source (PK_S), encrypt (SK^D). 3. Broadcast encrypted (SK^D) using PK_S as well as SK^E independently and cipher text of message to all. 4. At the receiver side decrypt SK^D using PK_S and SK^E, if both are equal then that key is meant for this receiver only otherwise not. 5. Decrypt the message using SK^D. Weaknesses of this protocol are: (a) number of messages exponentially increase with the group size. (b) CRT added extra computational cost (c) Session key SK^D or ciphertext-plaintext pair (SK^D, $E_{SK^E}\{SK^D\}$ are easy to impersonate as source or receiver.

4.2.2.3 ID BASED THRESHOLD KEY MANAGEMENT

4.2.2.3.1 D. Bonch and M. Franklin (DM)

This is the first identity based key generation technique using elliptic curve cryptosystem and Diffie Hellman key agreement and distribution protocol [66, 67]. The Diffie-Hellman algorithm used in this protocol is Bilinear Diffie-Hellman (BDH). Here, identity is chosen in an encrypted form. The protocol runs as: (i) choose a random number 'k' to generate two prime order groups G_1 and G_2. (ii) Pick a private/secret key 's' and generate public key K_{PB} = sP. Where, P is generator or primitive element in G such that $G_1 x G_2 \rightarrow G$. (iii) Choose a hash function 'H' from G_2. (iv) Shared public key is (q, G_1, G_2, K_{PB}, H, ID_{source}, n). This protocol is secure against chosen ciphertext attack. The security of this system is further suggested through El. Gamal cryptosystem. This is based on Diffie Hellman algorithm only. There is no discussion of timestamp in this protocol and thus it can be prone to parallel session attack.

4.2.2.3.2 F.R. Yu and H. Tang (FH)

This is an ID based threshold key management technique using intrusion detection system for MANET. In this approach node identity is based on gattins indicices[67]. These identities are calculated offline. This calculation helps in generation of the master key. This is a pair of public and private key. Some node holds the shares of master key and others can make a request for the private key to potential nodes. The protocol runs as: (i) a new node send a key request to potential nodes for private key forward the request through broadcasting to other potential nodes. (ii) A potential node having master private key forwards the request through broadcasting to other potential nodes. (iii) Security of these transmissions as well as system is checked through proper intrusion detection system. (iv) Computes gattins indices and keeps it unchanged for other node. (v) Collectively these gattin's indices help in computation of private key. This protocol enhances the computation power of the system and time complexity. It is much better than centralized key management techniques.

4.2.2.3.3 H. Deng, A Mukherjee and D. Agarwal (HAD)

This is another identity based threshold key management technique using Lagrange interpolation [67, 68]. This approach extended the Shamir's secret key cryptography to enhance the trust among authorities. Here, keys are not generated by trusted third party but it is computed from all nodes. Key's identification factors are computed either from IP address or node's unique identity. These are self configured or dynamically provided identities. Here, keys are exchanged on a one hop distance i.e. node1 will broadcast the key to neighbouring one hop distant nodes. Neighbouring node will further distribute keys to their neighbouring nodes that are one hop distant apart. Neighbouring nodes also sign the partial component of private key and send back to originating node as a feedback. Once initialization is over and there is need to further calculate new master key then partials of nodes are shuffled. This protocol provides high availability, confidentiality, authentication and non-repudiation.

4.2.2.3.4 Y. Zhang, W. Liu, W. Lou and Y. Fang (YWWY)

This is an identification based threshold key management and certificate less technique [67, 69]. The protocol runs as: (i) generate pairing parameters- which involves generation of prime numbers and hash function. (ii) Secret calculation- this is extended with Lagange coefficient with use of shamir's secret key cryptosystem. (iii) Generating identification based private/public key- random number hashed salts are used to identify nodes in the network. These hashes are generated using SHA-1. (iv) Updation: nodes can frequently join or leave. To handle such situation, polynomial based calculations are performed in galois field. As compared to centralized key mechanism the overhead and time is reduced to approximately 1/600th and 1/100th.

5 KEY CANCELLATION

Lifetime of a key depends upon type of key. Key can be classified as shared secret key, public key, private key, master key, random key, one time key etc. Although, it is most secure to refresh any type of key after regular intervals but some key need to be maintained for longer duration. For example, shared secret key is the key that need to be maintained for longer duration to share other keys. Other key's lifetime also depends upon type of connection. Some are connection oriented protocols (e.g. Transmission Control Protocol, Stream Control Transmission Protocol etc.) and other is connectionless (e.g. User Datagram Protocol, Congestion Control Datagram Protocol etc.). For connection oriented protocols, lifetime of the key is as long as the session's lifetime and for each new session there will be new session key. For connectionless protocols, lifetime of the key is renewed after each transmission. Lifetime of a key also depends upon certificate issued by certificate authority [11, 12]. Very frequent updating of key's lifetime also increases the traffic over network because of key management. Deciding the minimum and maximum duration is an open research issue.

6 CONCLUSIONS

In this work, several security related issues and challenges involved in key management protocols were analyzed and discussed. Detailed running scenarios of peer to peer and group communication protocols were analyzed and surveyed. Strengths and weaknesses of protocols were compared and tabled along with. In resource scared networks, the protocol's message size, number of keys required or generated per unit time, threshold etc. were important factors to be considered for comparisons. Security related issues in resource scared networks required further research attention.

7 FUTURE WORK

The future work of this paper is including the survey of other categories or protocols of key management i.e. centralized, decentralized, identity, threshold, distributed, multiple server, hierarchical, rekeying etc. We compare the above protocols for Mobile Ad Hoc Networks in terms of authentication, authorization, confidentiality and other security services.

REFERENCES

[1] C. Adam and S. Farrell."Internet X.509 public key infrastructure: Certificate management protocols." *Internet Request for Comments 2510*, 1999.

[2] C. Adam, P. Cain, D. Pinkas and R. Zuccherato,"Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", *Internet Request for Comments 3161*, August 2001.

[3] B. Ramsdell,"S/MIME Version 3 certificate handling." *Internet Request for Comments 2632*, 1999.

[4] J. Schaad, M. Myers,"Certificate Management over CMS (CMC): Transport Protocols", *Internet Request for Comments 5273*, June 2008.

[5] H. H. Chu, L. Qiao and K. Nahrstedt, "A Secure Multicast Protocol with copyright Protection", *ACM SIGCOMM Computer Communications Review,* 32(2):42-60, April 2002.

[6] D. Harkins, D. Carrel, "The Internet Key Exchange", *Internet Request for Comments 2409*, November 1998.

[7] D. Maughan, M. Schertler, M. Schneider and J. Turner, "Internet Security Association and Key Management Protocol", *Internet Request for Comments 2408*, November 1998.

[8] IETF RFC 2412, *The OAKLEY Key Determination Protocol,* November 1998.

[9] Hugo Krawczyk, "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", *IEEE Proceesings of Network and Distributed System Security* ,pp. 114-127 (FEB 1996).

[10] Henk C. A. van Tilborg, " Encyclopedia of Cryptography and Security", Springer-verlag, 2005.

[11] C. Adam and S. Farrell, "Internet X.509 public key infrastructure: Certificate management protocols." *Internet Request for Comments 2510*, 1999.

[12] B. Ramsdell, "S/MIME Version 3 certificate handling", Internet Request for Comments 2632.

[13] William Stalling, "Cryptography and Network Security: Principles and Practice", 5th Edition, 2011.

[14] R. Needham, M. Schroeder, "Using encryption for authentication in large networks of computers", *Communications of the ACM*, 21, Pages 993-999, (1978).

[15] D. Denning, G. Sacco, "Timestamps in key distribution protocols", *Communications of the ACM,* 24:8,(1981).

[16] Dave Otway, Owen Rees, "Efficient and Timely Mutual Authentication", ACM Operating System Review, 21(1), pp. 8-10, January 1987.

[17] C. Boyd, W. Mao, "On a Limitation of BAN Logic", *Advances in Cryptology-EUROCRYPT'93*, Lecture Notes in Computer Science 765, Tor Helleseth(Ed.), pp. 240-247, May 1993.

[18] W. Mao and C. Boyd, "Towards Formal Analysis of Security Protocols*", In Proceedings of the Computer Security Foundations Workshop VII*, pp.147, 158, IEEE Computer Society Press, Los Alamitos, California, 1993.

[19] C. Boyd and A. Mathuria, "Protocols for Authentication and Key Establishment", *Springer-Verlag,*1st Edition, 2003.

[20] M.O. Rabin, "Digitalized Signatures and Public Key Function as Intractable as Factorization", MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[21] U. Carlsen, "Optimal Privacy and Authentication on a Portable Communications System", *ACM Operating Systems Review*, 28(3), 1994, pp. 16-23.

[22] M. J. Beller , L. F. Chang and Y. Yacobi,"Privacy and Authentication on a Portable Communications System", *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 821-829, August 1993.

[23] V. Varadharajan and Y. Mu, "Design of Secure End-to-End Protocols for Mobile Systems", *Wireless 96 Conference,* Alberta,Canada, pp. 561-568.

[24] V. Varadharajan and Y. Mu, "On the Design of Security Protocols for Mobile Communications", *ACISP'96 Conference*, Springer-Verlag, 1996, pp. 134-145.

[25] C. Boyd and A. Mathuria, "Key establishment protocols for secure mobile communications: A selective survey", *Elsevier Computer Communication*, vol. 23, Issues 5-6, pp. 575-587, 1998.

[26] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithm", *IEEE Transaction on Information Theory*, vol. 31, pp. 469-472, 1985.

[27] M. Tatebayashi, N. Matsuzaki and D.B. Newman Jr., " Key Distribution Protocol for Digital Mobile Communication Systems" , *Advances in Cryptology Crypto'89,* Springer-Verlag, , pp. 324-333,1990.

[28] G. Lowe and A. W. Roscoe, "Using CSP to detect errors in the TMN protocol", *Software Engineering,* 23(10):659-669, 1997.

[29] Gustavus J. Simmons, "An impersonation-proof identity verification scheme", In Advances in Cryptology", *Proceedings of Crypto 87*, volume 293 of LNCS, pages 211-215. Springer-Verlag, 1988.

[30] Gustavus J. Simmons, "Cryptoanalysis and protocol failure", *Communications of the ACM,* 37(11):56-65, November 1994.

[31] Y. Yacobi and Z. Shmuely, "On key Distribution Systems", *Advances in Cryptolog-Crypto'89*, Springer-Verlag, pp. 344-355, 1989.

[32] C. S. Park, “On Certificate-Based Security Protocols for Wireless Mobile Communication Systems”, *IEEE Network*, pp. 50-55, Septem A. Aziz and W. Diffie, “Privacy and Authentication for Wireless Local Area Networks” , *IEEE Personal Communications,* vol. 1, pp. 25-31, 1994. September/October 1997

[33] A. Aziz and W. Diffie, “Privacy and Authentication for Wireless Local Area Networks” , *IEEE Personal Communications,* vol. 1, pp. 25-31, 1994.

[34] C. Meadows, “Formal Verification of Cryptographic Protocols: A Survey”, *Advances in Cryptology – ASIACRYPT ’94, vol. 917 of Lecture Notes in Computer Science,* pp. 135-150, Springer-Verlag, 1995.

[35] G. Horn and B. Prencel, “Authentication and Payment in Future Mobile Systems”, *Proceedings of ESORICS’ 98*, Springer-verlag, 1998.

[36] Keith Matrin and Chris Mitchell, “Evaluation of Authentication Protocol for Mobile Environment Value-added Services”, Draft, 1998.

[37] K. M. Martin, B. Preneel, C. J. Mitchell, H. J. Hitz, G. Horn, A. Poliakova, P. Howard, “ Secure billing for mobile information services in UMTS”, *Proceedings of ISEN’98.* 1998.

[38] T. Matsumato, Y. Takashima and H. Imai, “On Seeking Smart Public key Distribution Systems”, *In Transaction of the IECE of Japan*, E69, pp.99-106, 1986.

[39] R. Dutta and R. Barua, “Overview of Key Agreement Protocols”, Cryptology ePrint Archive, Report 2005/289.

[40] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, “ An efficient Protocol for Authenticated Key Agreement”, Technical Report CORR 98-05, Department of C&O, University of Waterloo, 1998.

[41] B. Kaliski, Contribution to ANSI X9F1 and IEEE P1363 working group, June 1996.

[42] M. Scott, “Authenticated ID-based key exchange and remote log-in with insecure token and PIN number”, Cryptology ePrint Archiive, Report 2002/164.

[43] N. P. Smart, “An Identity based Authenticated Key agreement Protocol Based on the Weil Pairing”, In Electronic Letters, 38, pp. 630-632, 2002.

[44] L. Chen and C. Kundla, “Identity Based Authenticated Key Agreement Protocols from Pairings”, 2002.

[45] K. Shim, “Efficient ID-based Authenticated Key Agreement Protocol Based on the Weil Pairing”, In Electronic Letters, 39(8), pp. 653-654, 2003.

[46] N. McCullagh and P.S.L.M. Barreto, “A New Two Party Identity Based Authenticated Key Agreement”, *In Proceedings of CT-RSA 2005*, LNCS 3376, pp. 262-274, Springer-Verlag, 2005.

[47] I. R. Jeong, J. Ketz and D.H. Lee, “One-Round Protocols for Two-Party Authenticated Key Exchang”, *In proceedings of ACNS 2004*, LNCS 3089, pp. 220-232, Springer-Verlag, 2004.

[48] Yair Amir, Yongdae Kim, Cristina Nita-Rotaru, and Gene Tsudik, “On the Performance of Group Key Agreement Protocols”, *ACM Transactions on Information and System Security”,* Vol. 7, No. 3, Pages 457-488, (August 2004).

[49] Paul Judge, Mostafa Ammar, “Security Issues and Solutions in Multicast Content Distribution: A Survey," IEEE Network Magazine, 2003.

[50] M.E. Hellman,”An Overview of Public Key Cryptography.” *IEEE Communications Society Magazine,* Vol. 16, Nov. 1978, pp. 24-32.

[51] Whit Diffie and Martin Hellman, “New Directions In Cryptography”, *IEEE Transactions on Information Theory,* IT-22(6):644-654, (November 1976).

[52] Michael Steiner, Gene Tsudik, and Michael Waidner, “ Diffie-Hellman Key Distribution Extended to Group Communication”, *ACM Conference on Computer and Communication Security,* Pages 31-37, (1996).

[53] Gluseppe Ateniese, Michael, and Gene Tsudik, “Authenticated Group Key Agreement and Friends” *International Conference on Computer and Communication Security,* Pages 17-26,(1998).

[54] H. Harney, C. Muckenhirn, “Group Key Management Protocol Architecture”, *Internet Request for Comments 2094*, July 1997.

[55] H. Harney, C. Muckenhirn, "Group Key Management Protocol Specification", *Internet Request for Comments 2093*, July 1997.

[56] H. Harney, U. Meth, A. Colegrove, "Group Secure Association Key Management Protocol", *Internet Request for Comments* 4535, June 2006.

[57] B. Weis, S. Rowles and T. Hardjono, " The Group Domain of Interpretation", *Internet Request for Comments 6407*, October 2011.

[58] M. Baugher, B. Weis, T. Hardjono, H. Harney, "The Group Domain of Interpretation", *Internet Request for Comments 3547*, July 2003.

[59] P. Hoffman, "Algorithm for Internet Key Exchange version 1 (IKEv1)", *Internet Request for Comments 4109*, May 2005.

[60] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol", *Internet Request for Comments 4306*, December 2005.

[61] T. H. Dunigan and C. Cao, "Group Key Management", Technical Report ORNL/TM-13470, 1998.

[62] M. Burmester and Y. Desmedt, "A Secure and scalable group key exchange system", In Information Processing Letters, 94(3), pp. 137-143, 2005.

[63] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system", *In proceedings of Eurocrypt*, LNCS 950, pp. 275-286, Springer-Verlag, 1995.

[64] R. Dutta and R. Barua, "Constant Round Dynamic Group Key Agreement", In proceedings of ISC 2005, LNCS, September 2005.

[65] G. H. Chion and W. T. Chen, "Secure Broadcast using Secure Lock", *IEEE Transaction on Software Engineering,* pp. 929-934, August 1989.

[66] D. Bonch and M. Franklin, "Identity-based encryption from weil pairing", *Advances in Cryptology-Crypto 2001*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

[67] J. V. D. Merwe, D. Dowoud and S. McDonald, "A Survey on Peer to Peer key management for Mobile Ad Hoc Networks", *ACM Computing Surveys*, vol. 39, No. 1, Article 1, April 2007.

[68] H. Deng, A. Mukherjee, D. Aggarwal, " Threshold and identity based key management and authentication for wireless ad hoc networks", *Proceedings of the international conference on information technology: Coding and Computing (ITCC's 04),* 2004.

[69] Y. Zhang, W. Liu, W. Lou and Y. Fang, " Securing mobile ad hoc networks with certificateless public keys", *IEEE Transaction on Dependable and Secure Computing*, vol. 3, pp. 386-399, 2006.

Mr. Adarsh Kumar is currently working as Senior Lecturer of Computer Science and Information Technology department at Jaypee Institute of Information Technology, Noida, INDIA, since September 2005. Mr. Kumar received his B.Tech (Computer Science) and M.Tech (Software Engineering) from Punjab Technical University and Thapar University, Patiala in June 2003 and July 2005 respectively. He is pursuing PhD in Computer Science from Jaypee Insttute of Information Technology, Noida, INDIA.

Dr. Alok Aggarwal is currently working as Assistant Professor in Computer Science and Information Technology department at Jaypee Institute of Information Technology, Noida, INDIA, since 2010.He is having work experience of fourteen years with a mix of software developer, research and teaching. He received his Bachelor, Master and PhD in computer science and engineering from kurukshetra university and IIT, Roorkee in 1995, 2001, 2010 respectively. He published three books and about forty research papers in different journals, conference proceedings etc.

Dr. Charu is currently working as Assistant Professor in Computer Science and Information Technology department at Jaypee Institute of Information Technology, Noida, INDIA, since 2011. She is having academic work experience of six years. She received her M.Tech and PhD in Computer Science and Engineering from Banasthali Vidyapith, Rajasthan and Kurukshetra University in 2005 and 2011 respectively. She has published about fifteen research papers in different national and international journals and conferences. She is member of various technical bodies like: ACM and ISTE.