Overview of Key Agreement Protocols

Ratna Dutta and Rana Barua

Stat-Math Unit
Indian Statistical Institute
203, B.T. Road, Kolkata
India 700108
e-mail:{ratna_r, rana}@isical.ac.in

Abstract

The emphasis of this paper is to focus on key agreement. To this aim, we address a self-contained, up-to-date presentation of key agreement protocols at high level. We have attempted to provide a brief but fairly complete survey of all these schemes.

1 Introduction

Key agreement is one of the fundamental cryptographic primitive after encryption and digital signature. Such protocols allow two or more parties to exchange information among themselves over an adversarially controlled insecure network and agree upon a common session key, which may be used for later secure communication among the parties. Thus, secure key agreement protocols serve as basic building block for constructing secure, complex, higher-level protocols.

The problem of designing efficient key agreement protocols, both in the two party and multi party (*i.e.* group) setting with lower computation and communication cost and round complexity have received much attention. The first pioneering work for key agreement is the Diffie-Hellman protocol given in their seminal paper [36] that invents the public key cryptography and revolutionizes the field of modern cryptography . However, the basic Diffie-Hellman protocol does not authenticate the two communication entities in the sense that an active adversary who has control over the channel can mount a man-in-the-middle attack to agree upon two separate keys with the users without the users being aware of this.

Authenticated Diffie-Hellman key agreement allows a pool of users within a large and completely insecure public network to establish a common secret key and ensures each user that no other principal aside from these specifically identified group of users can possibly learn the value of a particular secret key. This is *implicit key authentication* and the protocol is called authenticated key agreement (AK) protocol. Additionally, the authenticated key agreement protocols are designed to ensure the entities that they are indeed sharing this secret key with each other. This property is called *explicit key authentication* and the protocol is said to be authenticated key agreement with key confirmation (AKC) protocol. Over the years, a number of security properties have been seen to be important in key agreement protocols and different approaches have been developed to solve the problem. Standard key derivation, message authentication code (MAC), digital signature scheme *etc.* are basic tools used to authenticate a key agreement.

1.1 Model

Several variations of the Diffie-Hellman protocol and Joux protocol have been suggested to incorporate authentication and a trial and error approach has been adopted to provide informal security analysis of the key agreement protocols. However, most of these protocols were broken and some of these protocols have flaws that came to light years after its proposal. The main problem were that appropriate threat models and the goals of secure AK and AKC protocols lacked formal definitions. It is extremely important both to correctly define the security model and to prove the security of any proposed implementation in that model.

Bellare and Rogaway [14] first consider a formal treatment for provable security of protocols in two party setting. Adapting their work, Blake-Wilson, Johnson and Menezes [16] developed a security model for distributed computing and provided rigorous definitions of the goals of secure AK and AKC protocols within this model. They proposed concrete AK and AKC protocols that were proven to be secure within this framework in the random oracle model.

Bellare, Canetti and Krawczyk [9] introduced a modular approach to design and analyze key agreement protocols. They achieved the modularity by applying a protocol translation tool, called an authenticator/compiler to protocols proven secure in a much simplified adversarial setting where authentication of communication links is not required.

Based on these works, Bresson *et al.* [24, 25, 27] introduced further refinements and defined a sound formalization for the authenticated key agreement and provided provably secure protocols within this model. This is an important step and has been used to analyze key agreement protocols, both in the two party and multi party setting.

1.2 Survey on Previous Work

There are a very few key agreement protocols that have concrete security proofs against active adversaries in a well defined security model. We can classify the key agreement protocols into two categories:

- Certificate-based and
- ID-based.

The certificate-based protocols work by assuming that each entity has a static (long term) public/private Diffie-Hellman key pair, and each entity knows the public key of each other entity.

The static public keys are authenticated via certificates issued by a certifying authority (CA) by binding users' identities to static keys. When two entities wish to establish a session key, a pair of ephemeral (short term) Diffie-Hellman public keys are exchanged between them. The ephemeral and static keys are then combined in a way so as to obtain the agreed session key. The authenticity of the static keys provided by signature of CA assures that only the entities who posses the static keys are able to compute the session key. Thus the problem of authenticating the session key is replaced by the problem of authenticating the static public keys which is solved by using CA, a traditional approach based on a public key infrastructure (PKI).

However, in a certificate-based system, the participants must first verify the certificate of the user before using the public key of the user. Consequently, the system requires a large amount of computing time and storage. In 1984, Shamir [76] proposed the idea of ID-based cryptosystem where the identity information of a user functions as his public key. A private key generator (PKG), sometimes also referred to as key generation center (KGC), which is trusted by all users is responsible for the generation of users' corresponding private keys. Shamir gave a practical ID-based signature scheme and asked for ID-based encryption to simplify key management procedures in certificate-based public key infrastructure. A few key agreement protocols have been developed based on Diffie-Hellman and Shamir's key setup idea [46, 73]. Recently, Cocks [34] and Boneh and Franklin [18] have proposed two ID-based encryption schemes which potentially allow the replacement of a PKI with a system where ones identity becomes the public key and a trusted PKG helps to generate users' private key. Cocks' scheme is based on the Quadratic Residuosity problem, whilst that of Boneh and Franklin relies on the Weil Pairing. Shortly after that, many ID-based cryptographic protocols were developed (see [37] for a survey) based on pairings and is currently an area of very active research.

Two-party key agreement. Numerous Diffie-Hellman based AK and AKC protocols have been designed to add authentication (and key confirmation) to the Diffie-Hellman protocol; however, many have subsequently been found to have flaws. One of the well-known authenticated key agreement (AK) protocol in the Diffie-Hellman family is MTI protocol by Matsumoto, Takashima and Imai [62]. They designed three infinite families of key agreement protocols to provide implicit key authentication in the classical Diffie-Hellman key agreement protocol. However, the security analysis against active adversary is only heuristic. Law *et al.* [60] pointed out flaws in the protocols and presented an efficient authenticated key agreement protocol, often called MQV protocol. The security analysis of MQV protocol against active adversary is also heuristic. Both MTI and MQV family of protocols are certificate-based.

There are many ID-based key agreement protocols based on pairing. Scott [75] proposed an ID-based key agreement protocol where each user selects his own personal identity number (PIN) and a trusted PKG issues each user an individual secret associated with the identity of corresponding user. A value is calculated from both the individual secret and PIN number and placed inside a hardware token. The individual secret can be reconstructed from their memorized PIN number, identity and token.

Another ID-based authenticated key agreement was proposed by Smart [81] that combines the idea of Boneh and Franklin [18] with the tripartite Diffie-Hellman protocol of Joux [50]. The scheme uses weil pairing and requires all users involved in the key agreement to be clients of the same PKG. The protocol allows efficient ID-based escrow facility for sessions that enables low enforcement agencies to decrypt messages encrypted with the session keys, after having obtained the necessary warrants.

Chen and Kudla [32] developed an ID-based authenticated key agreement protocol more efficient than Smart's protocol [81]. They have suggested a mechanism to turn escrow off which can also be applied to Smart's protocol [81] (the escrow-free environment may be desirable for personal communications the users wish to keep confidential even from the PKG). They also provided a modification that allows key agreement between users under different PKGs.

None of the two party key agreement protocols by Scott [75], Smart [81] and Chen and Kudla [32] were broken, although heuristic arguments are adopted to prove their security against active adversary. Shim [77] presented an ID-based key agreement protocol. However, Sun and Heish [85] showed that Shim's key agreement protocol is insecure against the man-in-the-middle attack.

Another efficient ID-based authenticated key agreement protocol was proposed by McCullagh and Barreto [63] that can be used in either escrow or escrow-free mode. They also developed a scheme for key agreement between clients of different PKGs. The scheme is twice as efficient as the scheme in [32] without precomputation. Later, Xie [86] pointed out a flaw in it and removed this flaw by suggesting modifications for the protocol. Recently, Kwang and Choo [59] showed that both the scheme and its modified variant are not secure if the adversary is allowed to reveal non-partner players who had accepted the same session key.

Jeong *et al.* [49]proposed three simple single-round two-party key agreement protocols with detail security analysis in the security model of [24].

Three-party key agreement. In one of the breakthroughs in key agreement, Joux [50] proposed a three party single round key agreement protocol using pairings on elliptic curve. This was the first positive application of bilinear pairings in cryptography. However, just like Diffie-Hellman, Joux's protocol is unauthenticated and is susceptible to the man-in-the-middle attacks. This original scheme is not ID-based.

Al-Riyami and Paterson [1] proposed four tripartite authenticated key agreement protocols to provide implicit key authentication in Joux's protocol by incorporating certified public keys using ideas from MTI [62] and MQV [60] protocols. They argued heuristically that these protocols achieve some desirable security attributes against active adversary. Later, Shim [78] pointed out that the protocols are insecure against man-in-the-middle attack, key compromise impersonation attack and several known-key attacks.

In [67, 68], Nalla *et al.* proposed authenticated tripartite ID-based key agreement schemes that were broken by Chen [31] and Shim [79]. Zhang, Liu and Kim [88] developed an ID-based single round authenticated tripartite key agreement protocol, the authenticity of which is assured by Hess' [47] ID-based signature scheme and provided heuristic security analysis of the protocol against active adversary.

Group key agreement. Another direction of research on key agreement is to generalize the two party key agreement to multi party setting and consider the dynamic scenario where participants may join or leave a multi-cast group at any given time. As a result of the increased popularity of group oriented applications, the design of an efficient authenticated group key agreement protocol has recently received much attention in the literature.

A comprehensive treatment have been made to extend the two party (and three party) key agreement protocols to multi party setting. Notable solutions have been suggested by Ingemerson *et al.* [48], Burmester and Desmedt [28], Steiner *et al.* [83] and Becker and Willie [8]. All these works assume a passive (eavesdropping) adversary, and the last three provide rigorous proofs of security.

For practical applications, efficiency is a critical concern in designing group key agreement in addition to provable security. In particular, number of rounds may be crucial in an environment where number of group members are quite large and the group is dynamic. Handling dynamic membership changes get much attention to the current research community. A group key agreement scheme in a dynamic group must ensure that the session key is updated upon every membership change so that subsequent communication sessions are protected from leaving members and previous communication sessions are protected from joining members. Although this can be achieved by running any authenticated group key agreement protocol from scratch whenever group membership changes, alternative approaches to handle this dynamic membership more effectively would be clearly preferable in order to minimize cost of the rekeying operations associated with group updates.

The problem of key agreement in Dynamic Peer Groups (DPG) were studied by Steiner *et al.* [83]. They proposed a class of "generic n-party Diffie-Hellman protocols". Atenise *et al.* [3, 4] introduced authentication into the class of protocols and heuristically analyze their security against active adversary. Steiner *et al.* [84] consider a number of different scenario of group membership changes and introduced a complete key management suite CLIQUES studied specifically for DPGs which enable addition and exclusion of group members as well as refreshing of the keys. The security analysis of these schemes are heuristic against active adversaries. However, Pereira and Quisquater [74] have described a number of potential attacks, highlighting the need for ways to obtain greater assurance in the security of these protocols.

Bresson *et al.* [24, 25, 27] have recently given a formal security model for group authenticated key agreement. They provided the first provably secure protocols based on the protocols of Steiner *et al.* [83] for this setting which requires $O(n)$ rounds to establish a key among a group of n users. The initial works [25], [27] respectively consider the static and dynamic case, the security of both of which are in random oracle model following the formalized security model introduced by themselves under the computational Diffie-Hellman (CDH) assumption. They further refine in [24] the existing security model to incorporate major missing details, (*e.g.* strong corruption and concurrent sessions) and proposed an authenticated dynamic group Diffie-Hellman key agreement proven secure under the DDH assumption within this model. Their security result holds in the standard model instead of random oracle model.

Tree-based group key agreement. A different arrangement of participants for key agreement is to consider tree-based setting which requires $\log n$ rounds and has some computational advantages. Group key agreements in tree based setting are typically essential while the users are grouped into a hierarchical structure. The leaves of the tree denote individual users and each internal node corresponds to a user who acts as a representative for the set of users in the subtree rooted at that node. The representative users may have more computational resources than other users in the subtree.

There have been quite a number of tree based key agreement protocols. Kim, Perrig, Tsudik [56] extends the 2-party DH protocol to binary tree-based setting that yields a secure protocol suite, called Tree-based Group Diffie-Hellman (TGDH) which is both simple and fault tolerant. They have considered the dynamic scenario where a group of users can join or leave the group and introduced four protocols: Join, Leave, Merge and Partition. However, the security analysis against active adversary is completely heuristic.

Nalla and Reddy [66] extends Smart's ID-based two party single round authenticated protocol to multi-party ID-based key agreement using a binary tree structure and made heuristic arguments to prove that the protocol achieves some desirable security attributes against active adversary.

Barua *et al.* [7] presented a ternary tree based unauthenticated key agreement protocol by extending the basic Joux's protocol to multi-party setting and provide a proof of security against passive adversaries. They have further proposed in [39] a provably secure authenticated tree based group key agreement from the unauthenticated protocol of [7] and analyze the security in the model formalized by Bresson *et al.* [24]. Dutta and Barua [40] considered the dynamic case of the scheme in [39] that enables an user to join or leave the group at his desire retaining the tree structure with minimum key updates.

Constant round group key agreement. Recently, Katz and Yung [54] presented a detailed security analysis of a variant of two round unauthenticated group key agreement of Burmester and Desmedt [28](BD) in the standard model under decision Diffie-Hellman (DDH) assumption. They also provide a compiler construction, application of which makes the unauthenticated BD protocol to a provably secure three round authenticated group key agreement. Their security analysis is in the security model formalized by Bresson *et al.* [24]. The protocol achieves the nice property of forward secrecy where compromise of the long term secrets of one or more entities does not affect the security of previous session keys. However, this approach does not prevent attacks from malicious insiders as described in [52], *e.g.* existence of dishonest entities who deviates from the protocol – such as refusing to deliver messages or giving a valid signature on an incorrect message – can make the system insecure.

Choi, Hwang and Lee [33] extends the BD protocol in bilinear pairing-based setting, security of which relies on the hardness of CDH problem in the random oracle model. They have also constructed an ID-based authenticated group key agreement under Decision Hash Bilinear Diffie-

Hellman (DHBDH) assumption in the random oracle model. Both the protocols achieve forward secrecy.

Becker and Willie [8] introduced the octopus protocols and the cube protocols in order to minimize the number of exchanges. They studied lower bounds for the communication complexity of contributory key distribution and established lower bounds for the total number of messages, the total number of exchanges and the total number of necessary rounds. They derived a lower bound of only one round for multi-party group key agreement protocols and leave as an open question whether any group key agreement scheme can meet this bound.

Boyd and Nieto [21] proposed a constant round authenticated group key agreement with a security proof in the random oracle model that meets Becker-Willie's lower bound of one round. However, the protocol does not provide forward secrecy. Furthermore, the protocol is computationally asymmetric as it requires a "group leader" to perform $O(n)$ encryption and $O(n)$ communication each time a group key is established.

Another provably authenticated static group key agreement based on standard secret sharing techniques combined with ElGamal encryption scheme is proposed by Bresson and Catalano [22] using asynchronous network. The security is in the standard model under DDH assumption. However, this protocol is inefficient from point of view of the computation rate and suffers from a significant communication overhead both in terms of the number of messages sent by all members during the protocol execution and in terms of the number of bits communicated throughout the protocol execution.

Bresson, Chevassut, Essiari and Pointcheval [26] introduced a very efficient provably secure group key agreement in dynamic scenario suitable for restricted power devices and wireless environments. The protocol requires two rounds and is proven to be secure in the random oracle model. However, their exists a base station as a trustee. Later, Nam, Kim, Won [71] demonstrate certain flaws in the basic setup protocol of [26] and proposed a modified version of the scheme to remove these flaws.

Nam, Kim, Yang, Won [72] investigate the problem of contributory group key agreement over combined wired/wireless networks, consisting of arbitrary number of mobile devices with limited computational resources and general-purpose stationary high-performance computer. They have designed a 3-round generalized protocol which take advantage of the difference in computing power among users and uses a 2-round unauthenticated protocol, introduced by them, as a basic building block. Their 2-round basic protocol is proven to be secure against passive adversary under DDH assumption.

A communication-efficient dynamic group key agreement protocol well suited for a lossy and high-delay unbalanced network is developed by Nam *et al.* [70]. Their protocol enables conference key agreement in an environment that consists of mobile hosts with restricted computational resources and stationary hosts with relatively high computational capabilities. They analyze their scheme in the random oracle model and prove that it is secure under factoring assumption.

In Asiacrypt 2004, Kim *et al.* [58] proposed a very efficient constant round dynamic authenticated group key agreement protocol and provided a security analysis under CDH assumption in the random oracle model.

More recently, Dutta and Barua [41] presented a constant round group key agreement protocol (DB) which may be viewed as a variant of Burmester-Desmedt [28] protocol (BD) with considerably better efficiency and flexibility.

The rest of the chapter is organized as follows. Section 2 and Section 3 focus on two party key agreement. We include in Section 2 certain non-ID based two party key agreement protocols and in Section 3 some ID-based 2-party key agreement protocols. Section 4 deals with three party key agreement. We devote Section 5 for key agreement in multi party scenario. Section 6 concerns dynamic group key agreement. Finally, we conclude in Section 7.

2 Cryptographic Bilinear Maps

Let G_1, G_2 be two groups of the same prime order q. We view G_1 as an additive group and G_2 as a multiplicative group. A mapping $e : G_1 \times G_1 \rightarrow G_2$ satisfying the following properties is called a cryptographic bilinear map: (*Bilinearity*) $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$; (*Non-degeneracy*) if P is a generator of G_1, then $e(P, P)$ is a generator of G_2; and (*Computablity*) there exists an efficient algorithm to compute $e(P, Q)$. Modified Weil Pairing [18] and Tate Pairing [6] are examples of cryptographic bilinear maps.

3 Two Party Key Agreement

3.1 Diffie-Hellman Key Agreement

(Diffie, Hellman [36], 1976)

Diffie-Hellman (DH) proposed the first two-party single-round key agreement protocol in their seminal paper [36] that enables the users to compute a common key from a secret key and publicly exchanged information. No user is required to hold secret information before entering the protocol and each member makes an independent contribution to the common agreed key. This work invents the revolutionary concept of public-key cryptography and is the most striking development in the history of cryptography.

- Protocol Description :

  Setup : Let G be a finite multiplicative group of some large prime order q and g be a generator of G.

  Key Agreement : Assume that two entities A and B want to decide upon a common key. They perform the following steps.

  1. User A chooses a random $a \in Z_q^*$, computes $T_A = g^a$ and sends T_A to B.

2. User B chooses a random $b \in Z_q^*$, computes $T_B = g^b$ and sends T_B to A.
3. User A computes $K_A = T_B^a$ and similarly user B computes $K_B = T_A^b$.

If A and B execute the above steps honestly, they will agree upon a common key $K_{\mathsf{AB}} = K_A = K_B = g^{ab}$.

- Assumption :
  DLP is hard.

- Security :
  The protocol is unauthenticated in the sense that it is secure against passive adversaries. An active adversary can mount man-in-the-middle attack.

- Efficiency :

  Communication : Round required is 1 and group element (of G) sent per user is 1.

  Computation : Each user computes 2 exponentiations.

Note : Kim, Perrig, Tsudik [56] extends this 2-party DH protocol to binary tree-based setting that yields a secure protocol suite, called Tree-based Group Diffie-Hellman (TGDH) which is both simple and fault tolerant. They have considered the dynamic scenario where a group of users can join or leave the group and introduced four protocols: Join, Leave, Merge and Partition. However, the security analysis against active adversary is completely heuristic.

3.2 The MTI Key Agreement

(Matsumoto, Takashima, Imai [62], 1986)
In an attempt to provide implicit key authentication in the classical Diffie-Hellman [36] key agreement protocol, Matsumoto *et al.* [62] designed three infinite families of key agreement protocols. The MTI/A0 and MTI/C0 are two special cases of these families that are much studied in the literature. Here we describe these two protocols.

- Protocol Description :

  Setup : Let G be an elliptic curve additive group of some large prime order q and P be a generator of G. A certifying authority (CA) is used in the initial setup stage to provide certificates which bind users' identities to long-term secret keys. The certificate for entity A will be of the form $\mathsf{Cert}_A = (\mathcal{I}_A|W_A|P|\mathcal{S}_{\mathsf{CA}}(\mathcal{I}_A|W_A|P))$. Here $\mathcal{I}_A$ denotes the identity string of A, $|$ denotes concatenation of data items, $\mathcal{S}_{\mathsf{CA}}$ denotes the CA's signature and w_A, $W_A = w_A P$ are respectively the long term private key, public key of A.

  Key Agreement : Two entities A and B with respective certificates Cert_A, Cert_B, long term public/private key pairs (W_A, w_A) and (W_B, w_B) perform the following steps to decide upon a common agreed key.

(a) Protocol MTI/A0

1. User A generates $r_A \in_R Z_q^*$, computes $R_A = r_A P$ and sends (R_A, Cert_A) to B.
2. User B generates $r_B \in_R Z_q^*$, computes $R_B = r_B P$ and sends (R_B, Cert_B) to A.
3. User A computes $K_A = r_A W_B + w_A R_B$. Similarly user B computes $K_B = r_B W_A + w_B R_A$.

After an honest execution of the protocol, A and B will agree upon a common secret key $K_{\mathsf{AB}} = K_A = K_B = (w_A r_B + w_B r_A)P$.

(b) Protocol MTI/C0

1. User A generates $r_A \in_R Z_q^*$, computes $T_A = r_A W_B$ and sends T_A to B.
2. User B generates $r_B \in_R Z_q^*$, computes $T_B = r_B W_A$ and sends T_B to A.
3. User A computes $K_A = w_A^{-1} r_A T_B$ and similarly user B computes $K_B = w_B^{-1} r_B T_A$.

After an honest execution of the protocol, A and B decide upon the common secret key $K_{\mathsf{AB}} = K_A = K_B = r_A r_B P$.

Assumption :

DLP is hard.

Security :

It is heuristically argued that the protocols achieve implicit key authentication. Law *et al.* [60] pointed out flaws in the protocols. They proved that the MTI/A0 and MTI/C0 families of protocols respectively are vulnerable to the small subgroup attack and unknown key share attack and presented an efficient authenticated key agreement protocol, often called MQV protocol that withstands these attacks. MTI/A0 protocol does not provide forward secrecy since an adversary who learns w_A, w_B can compute all session keys established by A and B.

Efficiency :

Communication : Round required is 1, group element (of G) sent per user is 1.

Computation : In MTI/A0 protocol, each user computes 3 scalar multiplications and 1 addition in G. In MTI/C0 protocol, each user computes 2 scalar multiplications in G, 1 inverse in Z_q^* and 1 multiplication in Z_q^*.

3.3 The MQV Key Agreement

(Law, Menezes, Qu, Solinas, Vanstone [60], 1998)

- Protocol Description :

Setup : The setup is same as in 3.2 for MTI protocol. We denote by f the bit length of q, *i.e.* $f = \lfloor \log_2 q \rfloor + 1$. For a finite elliptic curve point $Q \in G$, $\overline{Q}$ is defined as follows. Let x be the x-coordinate of Q, and $\overline{x}$ be the integer obtained from the binary representation of x. Then $\overline{Q}$ is defined to be the integer $(\overline{x} \bmod 2^{\lceil f/2 \rceil}) + 2^{\lceil f/2 \rceil}$. Observe that $(\overline{Q} \bmod q) \neq 0$.

Key Agreement : Two entities A and B with respective certificates Cert_A, Cert_B, long term public/private key pairs (W_A, w_A) and (W_B, w_B) perform the following steps to decide upon a common agreed key.

1. User A generates $r_A \in_R Z_q^*$, computes $R_A = r_A P$ and sends (R_A, Cert_A) to B.
2. User B generates $r_B \in_R Z_q^*$, computes $R_B = r_B P$ and sends (R_B, Cert_B) to A.
3. User A computes $s_A = r_A + \overline{R}_A w_A \bmod q$ and $K_A = s_A(R_B + \overline{R}_B W_B)$.
4. User B computes $s_B = r_B + \overline{R}_B w_B \bmod q$ and $K_B = s_B(R_A + \overline{R}_A W_A)$.

If A and B follow the protocol, they will agree upon a common secret key $K_{\mathsf{AB}} = K_A = K_B = s_A s_B P = (r_A r_B + r_A w_B \overline{R}_B + r_B w_A \overline{R}_A + w_A w_B \overline{R}_A \overline{R}_B)P$.

Assumption :

DLP is hard.

Security :

The protocol possess the security attributes of known key security, forward secrecy, key compromise impersonation and key control. However, the security analysis is only heuristic. Later, Kaliski [53] observed that the the protocol does not possess the unknown key share attribute.

Efficiency :

Communication : Round required is 1, group element (of G) sent per user is 1.

Computation : Each user computes 3 scalar multiplications and 1 addition in G. Since the expression for $\overline{R}_A$ uses half the bits of the x-coordinate of R_A, the scalar multiplication $\overline{R}_A w_A$ can be done in half the time of a full scalar multiplication. Hence the work required by each entity is 2.5 full scalar multiplications. The on-line work required by each entity is only 1.5 scalar multiplications as $r_A P$ can be computed off-line. These result increased efficiency in key computation without affecting the security of the protocol.

3.4 Jeong, Katz and Lee's Key Agreement

(Jeong, Katz, Lee [49], 2004)

Jeong *et al.* [49] proposed three single-round key agreement schemes $\mathcal{TS}1$, $\mathcal{TS}2$, $\mathcal{TS}3$ which are simple variants of DH key agreement. They proved that the security of the scheme $\mathcal{TS}1$ and $\mathcal{TS}2$ are based on CDH assumption in the random oracle model whereas the security of the scheme $\mathcal{TS}3$ is based on DDH assumption in the standard model. The security analysis is in the security model as defined in [12, 14, 24]. The scheme $\mathcal{TS}1$ does not provide forward secrecy whilst both the schemes $\mathcal{TS}2$, $\mathcal{TS}3$ provide forward secrecy as well as key independence. We describe the scheme $\mathcal{TS}3$ below.

- Protocol Description :

Setup : Let $G = \langle g \rangle$ be a multiplicative group of some large prime order q and $H : \{0,1\}^* \rightarrow \{0,1\}^k$ ($k = |q|$) be a cryptographic hash function. We assume that $x_i, y_i = g^{x_i}$ are respectively the private, public key pair of an entity P_i. We also assume that the entities can be ordered by their names (*e.g.* lexicographically) and write $P_i < P_j$ to denote this ordering.

Key Agreement: Assume that two entities P_i, P_j wants to establish a session key and $P_i < P_j$. They perform the following steps.

1. User P_i first computes $K_{i,j} = y_j^{x_i}$ that it will use as a key for a message authentication code ($K_{i,j}$ may need to be hashed before being used). Then P_i chooses an ephemeral key $\alpha_i \in Z_q^*$ at random, computes a tag $\tau_i \leftarrow \mathsf{Mac}_{k_{i,j}}(i|j|g^{\alpha_i})$ and sends $g^{\alpha_i}|\tau_i$ to P_j.
2. Similarly, user P_j computes a key $K_{j,i} = y_i^{x_j}$ for a message authentication code, chooses an ephemeral key $\alpha_j \in Z_q^*$ at random, computes a tag $\tau_j \leftarrow \mathsf{Mac}_{k_{j,i}}(j|i|g^{\alpha_j})$ and sends $g^{\alpha_j}|\tau_j$ to P_i.
3. User P_i, on receiving the message, verifies the tag using $k_{i,j}$. If verification fails, no session key is computed. Otherwise, P_i computes a session key $K_i = (g^{\alpha_j})^{\alpha_i}$ with session identifier $\mathsf{sid}_i = g^{\alpha_i}|\tau_i|g^{\alpha_j}|\tau_j$.
4. Similarly, P_j verifies the tag of the received message using $k_{j,i}$. If verification fails, no session key is computed. Otherwise, P_j computes a session key $K_j = (g^{\alpha_i})^{\alpha_j}$ with session identifier $\mathsf{sid}_j = g^{\alpha_j}|\tau_j|g^{\alpha_i}|\tau_i$.

If P_i, P_j follow the above steps, they will agree upon a common secret key $K_{i,j} = K_i = K_j = g^{\alpha_i \alpha_j}$ with a common session identifier $\mathsf{sid}_i = \mathsf{sid}_j$.

- Assumption :
  DDH problem is hard and the message authentication code (MAC) used in the protocol is strongly unforgeable.

- Security :
  The protocol is proven to be secure in the standard model using the security model as defined in [12, 24] instead of using heuristic arguments. The protocol provides forward secrecy and key independence assuming that the MAC is secure and DDH problem is hard.

- Efficiency :

  Communication : Round required is 1 and total message size communicated per user is $|q| + |\mathsf{Mac}|$.

  Computation : Each user computes 3 modular exponentiations and 1 MAC computation.

4 Two Party ID-Based Key Agreement

4.1 Smart's Key Agreement

(Smart [81], 2002)

Smart proposed an ID-based authenticated key agreement protocol by combining the ideas from [18, 50, 60]. The scheme requires that all users involved in the key agreement are clients of the same Private Key Generator (PKG).

- Protocol Description :

  Setup : Suppose G_1, G_2, e are same as defined in Definition 2 of cryptographic bilinear maps. The PKG chooses a secret key $s \in Z_q^*$ and sets $P_{pub} = sP$. Let $H_1 : \{0,1\}^* \rightarrow G_1^*$ be a Map-to-point hash function. The master key of PKG is s and the global public key is P_{pub}. The system parameters and master public key are distributed to the users through a secure authenticated channel.

  Extract : Given a public identity $\mathsf{ID} \in \{0,1\}^*$, the PKG computes the public key $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathsf{G}_1$ and generates the associated private key $S_{\mathsf{ID}} = sQ_{\mathsf{ID}}$.

  Key Agreement : Let two users A and B with public keys respective $Q_A = H_1(\mathsf{ID}_A)$ and $Q_B = H_1(\mathsf{ID}_B)$ decide to agree upon a common secret key. They perform the following operations.

  1. User A chooses an ephemeral key $a \in_R Z_q^*$, computes $T_A = aP$ and sends T_A to B.
  2. User B chooses an ephemeral key $b \in_R Z_q^*$, computes $T_B = bP$ and sends T_B to A.
  3. User A computes $K_A = e(aQ_B, P_{pub})\ e(S_A, T_B)$ where $S_A = sQ_A$ is the long term secret key of A sent by the PKG on submitting A's public identity.
  4. User B computes $K_B = e(bQ_A, P_{pub})\ e(S_B, T_A)$ where $S_B = sQ_B$ is the long term secret key of B sent by the PKG on submitting B's public identity.
  5. After an honest execution of the above steps, both A and B will share the common agreed key $K_{\mathsf{AB}} = K_A = K_B = e(aQ_B + bQ_A, P_{pub})$.

- Assumption :
  The classical DLP and CDH problem are hard.

- Security :
  It is heuristically argued that the protocol posses the security properties: mutual implicit key authentication, known key security, partial forward secrecy, imperfect key control, key compromise impersonation and unknown key-share resilience. Smart also proposed in the paper an ID-based authenticated key agreement protocol with key confirmation property. Shim [77] discussed that Smart's protocol does not posses perfect forward secrecy and proposed a modified scheme which in turn is proven to be insecure against man-in-the-middle attack by Sun and Hsieh [85].

- Efficiency :

  Communication : Round required is 1, group element (of G_1) sent per user is 1.

  Computation : Each user computes 1 scalar multiplication in G_1, 2 pairing computations, 1 multiplication in G_2 and 1 Map-to-point hash operation. Additionally, the PKG requires to compute 1 Map-to-point hash operation, 1 scalar multiplication per client and also 1 scalar multiplication to generate P_{pub}.

Note : The protocol allows efficient ID-based escrow facility for sessions that enables law enforcement agencies to decrypt messages encrypted with the session keys, after having obtained the necessary warrants. Nalla and Reddy [66] extends this protocol to multi-party ID-based key agreement using a binary tree structure and made heuristic arguments to prove that the protocol achieves some desirable security attributes.

4.2 Scott's Key Agreement

(Scott [75], 2002)

Scott proposed an ID-based scheme where each user selects their own PIN number and a trusted PKG issues each user an individual secret associated with the identity of corresponding user. A value is calculated from both the individual secret and PIN number and placed inside a hardware token. The individual secret can be reconstructed from their memorizes PIN, identity and token.

- Protocol Description :

  Setup : Same as in section 4.1 for Smart's protocol.

  Extract : For individual clients to register with the PKG, they must prove their identity. Given the public identity $\mathsf{ID}_A \in \{0,1\}^*$ of an user A, the PKG computes the public key $Q_A = H_1(\mathsf{ID}_A) \in G_1$, generates the associated private key $S_A = sQ_A$. After authenticating himself, the user A receives S_A, calculates $\alpha_A Q_A$ where α_A is the desired secret PIN of A, subtracts the two and places the value $(s-\alpha_A)Q_A$ inside a hardware token. A memorizes α_A and then discard the secret S_A which it can reconstruct using the token, PIN and identity.

  Key Agreement : Let two users A, B with respective public keys Q_A, Q_B want to agree upon a common session key. They executes the following steps.

  1. A picks $a \in_R Z_q^*$, computes $T_A = e((s-\alpha_A)Q_A + \alpha_A Q_A, Q_B)^a$ and sends T_A to B.
  2. B picks $b \in_R Z_q^*$, computes $T_B = e((s-\alpha_B)Q_B + \alpha_B Q_B, Q_A)^b$ and sends T_B to A.
  3. A computes $K_A = T_B^a$ and similarly user B computes $K_B = T_A^b$.

  If both A and B follow the protocol, they will agree upon a common key $K_{\mathsf{AB}} = K_A = K_B = e(Q_A, Q_B)^{sab}$. (Scott used Tate pairing of order r in their protocol and the ephemeral keys a, b chosen respectively by A, B are less than r.)

- Assumption :
  The classical DLP and CDH problem are hard.
- Security :
  The author informally argued that the scheme is secure against impersonation attack.
- Efficiency :

  Communication : Round required is 1, group element (of G_2) sent per user is 1.

  Computation : Each user computes 1 scalar multiplication in G_1, 1 pairing computation, 2 exponentiation in G_2, 1 Map-to-point hash operation and 1 subtraction in G_1. Additionally, the PKG requires to compute 1 Map-to-point hash operation and 1 scalar multiplication per client and also 1 scalar multiplication to generate P_{pub}.

4.3 Chen and Kudla's Key Agreement

(Chen, Kudla [32], 2002)

In this work, Chen and Kudla presented an identity-based authenticated key agreement protocol more efficient than Smart's protocol [81] and analyzed the security using formal security model of [12, 13]. They have suggested a mechanism to turn escrow off which can also be applied to Smart's protocol [81] (the escrow-free environment may be desirable for personal communications the users wish to keep confidential even from the PKG). They also provided another modification that allows key agreement between users under different PKGs.

- Protocol Description :

  Setup : Same as for Smart's protocol in section 4.1.

  Extract : Same as in section 4.1

  Key Agreement : Users A, B with public keys Q_A, Q_B respectively performs the following steps to decide upon a common secret key.

  1. User A chooses an ephemeral key $a \in_R Z_q^*$, computes $T_A = aQ_A$ and sends T_A to B.
  2. User B chooses an ephemeral key $b \in_R Z_q^*$, computes $T_B = bQ_B$ and sends T_B to A.
  3. User A computes $K_A = e(S_A, T_B + aQ_B)$ and similarly user B computes $K_B = e(S_B, T_A + bQ_A)$.

  After an honest execution of the protocol, A and B agree upon a common session key $K_{\mathsf{AB}} = K_A = K_B = e(Q_A, Q_B)^{s(a+b)}$.

- Assumption :
  BDH problem is hard.

- Security :
  The authors adopt the security model of [12, 13] and prove the security of their protocol in the random oracle model assuming that the adversary makes no Reveal query. It is heuristically argued that the protocol achieves the security properties: partial forward secrecy, imperfect key control, unknown key share resilience and key compromise impersonation.

- Efficiency :

  Communication : Round required is 1, group element (of G_1) sent per user is 1.

  Computation : Each user computes 2 scalar multiplications in G_1, 1 pairing computation, 2 exponentiation in G_2 and 2 Map-to-point hash operation. Additionally, the PKG requires to compute 1 Map-to-point hash operation and 1 scalar multiplication per user and 1 scalar multiplication to generate P_{pub}.
  Clearly, the scheme is efficient compared to Smart's protocol [81].

4.4 McCullagh and Barreto's Key Agreement

(McCullagh, Barreto [63], 2004)

McCullagh and Barreto [63] designed an efficient ID-based authenticated key agreement protocol that can be used in either escrow or escrow-free mode and also a scheme for key agreement between clients of different PKGs. The scheme is twice as efficient as the scheme in [32] without precomputation. We describe below the key agreement scheme with escrow.

- Protocol Description :

  Setup : The setup is same as in Smart's protocol in section 4.1.

  Extract : The PKG verifies the on line public identity ID_A of A and computes $a = H_1(\mathsf{ID}_A)$ and $Q_A = (a+s)P$. Q_A is the public key of A, which can also be computed as $aP + P_{pub}$. The PKG then calculates A's private key as $S_A = (a+s)^{-1}P$.

  Key Agreement : Two entities A, B perform the following steps to agree upon a common key.

  1. User A chooses an ephemeral key $x_a \in_R Z_q^*$, computes $T_A = x_a Q_B$ and sends T_A to B.
  2. User B chooses an ephemeral key $x_b \in_R Z_q^*$, computes $T_B = x_b Q_A$ and sends T_B to A.
  3. User A computes $K_A = e(T_B, S_A)^{x_a}$ and similarly user B computes $K_B = e(T_A, S_B)^{x_b}$.

  If A and B follow the protocol, they will the same shared secret key $K_{\mathsf{AB}} = K_A = K_B = e(P, P)^{x_a x_b}$.

- Assumption :
  BDHI problem is hard.

- Security :
  The security analysis of the protocol is in the security model of [12, 13] assuming that the adversary makes no Reveal query and using random hash oracle. Heuristic arguments show that the protocol achieves the security properties: known key security, key compromise impersonation, forward secrecy, unknown key share resilience and key control. Later, Xie [86] pointed out that a malicious adversary can successfully launch a key compromise attack. He removed this flaw by suggesting modifications for the protocol. Recently, Kwang and Choo [59] showed that both the scheme and its modified variant are not secure if the adversary is allowed to reveal non-partner players who had accepted the same session key.

- Efficiency :

  Communication : Round required is 1 and group element (of G_1) sent per user is 1.

  Computation : Each user computes 1 scalar multiplication in G_1, 1 pairing computation, 1 exponentiation in G_2 and 1 hash (H_1) operation. Additionally, the PKG requires to compute 1 hash operation and 1 scalar multiplication per user and also 1 scalar multiplication to generate P_{pub}.
  The scheme is efficient than the schemes in [32, 81].

5 Three Party Key Agreement

5.1 Joux Key Agreement

(Joux [50], 2000)

Joux introduced a very simple and elegant tripartite key agreement protocol which makes use of bilinear pairing on elliptic curves that requires just one broadcast per entity. This was a major breakthrough in key agreement and was the first positive application of pairing in cryptography. Following this work, a number of pairing-based protocols were proposed.

- Setup : Let G_1, G_2, e be as defined in Smart's protocol in section 4.1. P is a generator of the additive group G_1 of order q, G_2 is a multiplicative group of same order q and e is the bilinear map from $G_1 \times G_1 \rightarrow G_2$.

- Protocol Description :

  Key Agreement : Consider three entities A, B, C decide to agree upon a common secret key. They perform the following steps.

  1. User A chooses $a \in_R Z_q^*$, computes aP and sends aP to both B and C.
  2. User B chooses $b \in_R Z_q^*$, computes bP and sends bP to both A and C.
  3. User C chooses $c \in_R Z_q^*$, computes cP and sends cP to both A and B.

4. User A computes $K_A = e(bP, cP)^a$, user B computes $K_B = e(aP, cP)^b$ and user C computes $K_C = e(aP, bP)^c$.

If A, B, C execute the above steps honestly, then they will agree upon a common key $K_{\mathsf{ABC}} = K_A = K_B = K_c = e(P, P)^{abc}$.

- Assumption :
  BDH problem is hard.

- Security :
  Joux's protocol is unauthenticated in the sense that it is secure against a passive adversary and suffers from the man-in-the-middle attack in presence of an active adversary.

- Efficiency :

  Communication : Round required is 1, group element (of G_1) sent per entity is 1.

  Computation : Each entity computes 1 scalar multiplication in G_1, 1 pairing computation and 1 exponentiation in G_2.

Note : Al-Riyami and Paterson [1] proposed four tripartite authenticated key agreement protocols to provide implicit key authentication in Joux's protocol by incorporating certified public keys using ideas from MTI [62] and MQV [60] protocols. They argued heuristically that these protocols achieve some desirable security attributes. Later, Shim [78] pointed out that the protocols are insecure against man-in-the-middle attack, key compromise impersonation attack and several known-key attacks.

5.2 Zhang, Liu and Kim's ID-Based Key Agreement

(Zhang, Liu, Kim [88], 2002)

In this work, an ID-based one round authenticated tripartite key agreement protocol is proposed by incorporating Hess' [47] ID-based signature.

- Protocol Description :

  Setup : Let G_1, G_2 be two groups of some large prime order q. We take G_1 to be an additive group and G_2 to be a multiplicative group. It is assumed that DL problem is hard in both G_1, G_2. Let P be a generator of G_1. We also consider a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. The PKG chooses a secret key $s \in Z_q^*$ and sets $P_{pub} = sP$. Let $H_1 : \{0,1\}^* \rightarrow G_1^*$ be a Map-to-point hash function. We also consider a cryptographic hash function $H : G_1 \rightarrow Z_q^*$. The master key of PKG is s and the global public key is P_{pub}.

  Extract : Given a public identity $\mathsf{ID} \in \{0,1\}^*$, the PKG computes the public key $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathsf{G}_1$ and generates the associated private key $S_{\mathsf{ID}} = sQ_{\mathsf{ID}}$.

Key Agreement : Three entities A, B, C with respective static (or long term) public keys $Q_A = H_1(\mathsf{ID}_A)$, $Q_B = H_1(\mathsf{ID}_B)$, $Q_C = H_1(\mathsf{ID}_C)$ and respective static (or long term) private keys $S_A = sQ_A$, $S_B = sQ_B$, $S_C = sQ_C$ perform the following steps to agree upon a common key.

1. User A chooses an ephemeral key $a \in Z_q^*$ at random, computes $P_A = aP$, $T_A = H(P_A)S_A + aP$ and sends (P_A, T_A) to both B and C.
2. User B chooses an ephemeral key $b \in Z_q^*$ at random, computes $P_B = bP$, $T_B = H(P_B)S_B + bP$ and sends (P_B, T_B) to both A and C.
3. User C chooses an ephemeral key $c \in Z_q^*$ at random, computes $P_C = cP$, $T_C = H(P_C)S_C + cP$ and sends (P_C, T_C) to both A and B.
4. User A verifies $e(T_B + T_C, P) = e(H(P_B)Q_B + H(P_C)Q_C, P_{pub})\ e(P_B, P_B)\ e(P_C, P_C)$ and computes $K_A = e(P_B, P_C)^a$ only if the verification succeeds.
5. User B verifies $e(T_A + T_C, P) = e(H(P_A)Q_A + H(P_C)Q_C, P_{pub})\ e(P_A, P_A)\ e(P_C, P_C)$ and computes $K_B = e(P_A, P_C)^b$ only if the verification succeeds.
6. User C verifies $e(T_B + T_A, P) = e(H(P_B)Q_B + H(P_A)Q_A, P_{pub})\ e(P_B, P_B)\ e(P_A, P_A)$ and computes $K_C = e(P_B, P_A)^c$ only if the verification succeeds.
   If the entities A, B, C follow the protocol, they will agree upon a common session key $K_{\mathsf{ABC}} = K_A = K_B = K_C = e(P, P)^{abc}$.

- Assumption :
  BDH problem and Weak-DH are hard (Hess' ID-based signature is secure under Weak-DH assumption).

- Security :
  Heuristic arguments shows that the protocol achieves the security attributes: implicit key authentication, known session key security, perfect forward secrecy, no key compromise impersonation, no unknown key share and no key control assuming that the underlying signature scheme (Hess's signature) is secure and BDH problem is hard.

- Efficiency :

  Communication : Round required is 1 and group elements (of G_1) sent per user is 2.

  Computation : Each user computes 5 scalar multiplications in G_1, 5 pairing computations, 2 multiplications in G_2, 2 Map-to-point hash operation (H_1) and 2 hash function (H) evaluation. Additionally, the PKG requires to compute 1 Map-to-point hash operation and 1 scalar multiplication per user and also 1 scalar multiplication to generate P_{pub}.

Note : Barua *et al.* [7] presented a ternary tree based unauthenticated key agreement protocol by extending the basic Joux's protocol to multi-party setting and provide a proof of security against passive adversaries. They have further proposed in [39] a provably secure authenticated tree based group key agreement from the unauthenticated protocol of [7] and analyze the security in the model formalized by Bresson *et al.* [24]. Dutta and Barua [40] considered the dynamic case of the scheme in [39] that enables an user to join or leave the group at his desire retaining the tree structure with minimum key updates.

6 Multi Party Key Agreement

6.1 Ingemarsson, Tang and Wong's Group Key Agreement

(Ingemarsson, Tang, Wong [48], 1982)

Since the publication of 2-party DH key exchange in 1976, various solutions have been proposed to extend DH key exchange to multi-party key distribution. Notable solutions have been proposed by Ingemarsson *et al.* [48] in 1982. We describe one of the families of protocols proposed by them.

- Protocol Description :

  Setup : Let $G = \langle g \rangle$ be a multiplicative group of some large prime order q.

  Key Agreement : Assume that n participants $U_1, \ldots, U_n$ want to agree upon a common key. The participants must be arranged in a logical ring. In a given round, every participant raises the previously received intermediate key value to the power of its own exponent and forwards the result to the next participant. The actual protocol is as follows.

  1. In round 1, user U_i, $1 \leq i \leq n$, chooses a random $x_i \in Z_q^*$, computes g^{x_i} and forwards it to $U_{(i+1) \bmod n}$.
  2. In round $k \in [1, n-1]$, user U_i, $1 \leq i \leq n$ computes $g^{\Pi\{x_j | j \in [(i-k) \bmod n, i]\}}$ and forwards it to $U_{(i+1) \bmod n}$.
  3. After $n-1$ rounds, all user agree upon a common session key $K = g^{x_1 x_2 \ldots x_n}$.

- Assumption :
  DDH problem is hard.

- Security :
  This protocol falls into the class of "natural" extensions of DH 2-party protocol and is secure against a passive adversary.

- Efficiency :

  Communication : Rounds required are $n-1$, messages sent per user are $n-1$.

  Computation : Each user computes n modular exponentiations.

6.2 Burmester and Desmedt Group Key Agreement

(Burmester, Desmedt [28], 1994)

Burmester and Desmedt presented a much more efficient key agreement protocol (BD) in group setting that requires only two rounds.

- Protocol Description :

  Setup : Let $G = \langle g \rangle$ be a multiplicative group of some large prime order q.

  Key Agreement : When n users $U_1, \ldots, U_n$ wish to establish a session key, they proceed as follows where the indices are taken modulo n so that user U_0 is U_n and user U_{n+1} is U_1.

  1. Each user U_i, $1 \leq i \leq n$, choose a random $x_i \in Z_q^*$ and broadcasts $Z_i = g^{x_i}$.
  2. Each user U_i broadcasts $X_i = \left(\frac{Z_{i+1}}{Z_{i-1}}\right)^{x_i}$.
  3. Each user U_i computes their session key as

  $$K_i = (Z_{i-1})^{nx_i}\ X_i^{n-1}\ X_{i-1}^{n-2}\ \ldots\ X_{i+n-2} \bmod q.$$

  If all the users U_i for $1 \leq i \leq n$ follow the above steps, they will agree upon the same key $K = g^{x_1x_2+x_2x_3+\cdots+x_nx_1}$.

- Assumption :
  DDH problem is hard.

- Security :
  The protocol is unauthenticated in the sense that it is secure against passive adversaries. However, the authors have not proposed any authentication method and any clear security proof. In [54], Katz and Yung investigate the security of a variant of BD protocol for unauthenticated group key agreement in detail and proposed a scalable compiler which transforms a secure unauthenticated group key agreement protocol into a secure authenticated group key agreement protocol preserving forward secrecy of the original protocol. They adopt the security model as formalized by Bresson *et al* [24] for security analysis.

- Efficiency :

  Communication : Rounds required are 2, messages sent per user are 2.

  Computation : Each user computes at most 3 (full length) modular exponentiations and $(\frac{n^2}{2} + \frac{3n}{2} - 3)$ modular multiplications.
  Katz-Yung modification to BD protocol adds one more round and performs additionally 2 signature generations and $(2n - 2)$ signature verifications.

Note 1 : Choi, Hwang and Lee [33] proposed a group key agreement protocol which is a bilinear version of the BD protocol and security of which relies on the hardness of CDH problem in the random oracle model. They have also constructed an ID-based authenticated group key agreement under Decision Hash Bilinear Diffie-Hellman (DHBDH) assumption in the random oracle model. Both the protocols achieve forward secrecy.

Note 2 : More recently, Dutta and Barua [41] presented a constant round group key agreement protocol (DB) which may be viewed as a variant of Burmester-Desmedt [28] protocol (BD) with considerably better efficiency and flexibility. Although the DB protocol is similar to BD protocol, there are subtle differences between them.

1. Key computation in DB protocol is different and is more efficiently done than in BD protocol.
2. Number of rounds, point-to-point communication, signature verifications require in DB protocol are less as compared to BD protocol and number of modul ar multiplications reduces from $O(n^2)$ to $O(n)$ with the same number of modular exponentiations.
3. DB protocol is more flexible than BD protocol in the sense that DB protocol is dynamic.
4. DB protocol has the ability to detect the presence of corrupted group members, although one can not detect who among the group members are behaving i mproperly.

The emphasis of this work is to achieve provable security of the scheme DB under DDH assumption. A concrete security analysis of this protocol is provided against active adversary in the standard security model of Bresson *et al.* [24] adapting Katz-Yung [54] technique. The protocol is forward secure, efficient and fully symmetric.

6.3 Steiner, Tsudik and Waidner's Group Key Agreement

(Steiner, Tsudik, Waidner [83], 1996)

Steiner *et al.* [83] defined a class of "generic n-party DH protocols" for which they have showed that security against passive adversaries is based on the intractability of the DDH problem. We describe three Group Diffie-Hellman (GDH) protocols: GDH.1, GDH.2 and GDH.3 introduced by them.

- Protocol Description :

  Setup : Let $G = \langle g \rangle$ be a cyclic group of a large prime order q.

  Key Agreement : Let users $U_1, \ldots U_n$ wishing to agree upon a common session key. They proceed the following steps.

(a) Protocol GDH.1

The protocol executes in $2(n-1)$ rounds and consists of two stages: up flow and down flow. The purpose of up flow stage is to collect contributions from all group members. The actual protocol execution is as follows.

1. *(Up flow)* In round i, $1 \leq i \leq n-1$, user U_i selects a random $x_i \in Z_q^*$ and sends $\{g^{\Pi(x_k|k \in [1,j])} \, | j \in [1,i]\}$ to U_{i+1}.
2. *(Down flow)* In round $(n-1+i)$, $i \in [1, n-1]$, user U_{n-i} sends $\{g^{\Pi(x_k|k \notin [i,j])} \, | j \in [1,i]\}$ to user U_{n-i+1}.

If all the users follow the above steps, they will agree upon a common secret key $K = g^{x_1 x_2 \ldots x_n}$.

(b) Protocol GDH.2

The protocol executes in n rounds and consists of two stages. In the first stage ($n-1$ rounds) contributions are collected from individual group members and then, in the second stage (n-th round), the group keying material is broad-casted. The actual protocol is as follows.

1. *(Up flow)* In round i, $1 \leq i \leq n$, user U_i selects a random $x_i \in Z_q^*$ and sends $\{g^{\frac{x_1x_2\ldots x_i}{x_j}} | j \in [1, j]\}$ and $g^{x_1x_2\ldots x_i}$ to U_{i+1}.
2. *(Broadcast)* In round n, U_n selects a random $x_n \in Z_q^*$ and broadcasts $\{g^{\frac{x_1x_2\ldots x_n}{x_i}} | i \in [1, n[\}$ to the rest of the users.

If all the users follow the protocol, they will agree upon a common secret key $K = g^{x_1x_2\ldots x_n}$.

(c) Protocol GDH.3

This protocol consists of four stages. The protocol execution among n users $U_1, \ldots, U_n$ requires $n+1$ rounds and proceeds as follows.

1. *(Up flow)* In the first stage, user U_i in the i-th round, $1 \leq i \leq n-2$, selects a random $x_i \in Z_q^*$, computes $g^{\Pi\{x_k | k \in [1,i]\}}$ and sends it to U_{i+1}.
2. *(Broadcast)* After processing the up flow message, U_{n-1} obtains $g^{\Pi\{x_k | k \in [1,n-1]\}}$ and broadcasts this value in the second stage to the rest of the participants. This is $n-1$-th round.
3. *(Response)* In the n-th round, each user U_i $(i \neq n)$, factors out its own exponent x_i and forwards the result to U_n. Thus user U_n receives the value $g^{\Pi\{x_k | k \in [1,n-1] \wedge k \neq i\}}$ in the n-th round. Note that factoring out x_i by U_i requires to compute its inverse x_i^{-1} which is always possible since the underlying group is of prime order.
4. *(Broadcast)* In the final stage, U_n collects all inputs from the previous stage, raises every one of them to the power x_n and broadcasts the resulting $n-1$ values $\{g^{\Pi\{x_k | k \in [1,n] \wedge k \neq i\}} | i \in [1, n-1]\}$ to the rest of the users.

Note that every user U_i now as a value of the form $g^{\Pi\{x_k | k \in [1,n] \wedge k \neq i\}}$ and can easily generate the intended group key $K = g^{x_1x_2\ldots x_n}$.

- Assumption :
  DDH problem is hard.
- Security :
  The above class of protocols are proven to be secure against passive adversaries under DDH assumption. Ateniese, Steiner, Tsudik [3, 4] studied these protocols for Dynamic Peer Groups (DPG) and provided an authentication mechanism to protocol GDH.2 with key confirmation and integrity. They adopt heuristic arguments to show that their authenticated protocols

achieve certain desirable security attributes. The problem of key agreement in DPG is also studied by Steiner, Tsudik, Waidner [84]. They considered all group key agreement operations (member addition, member exclusion, mass join, mass leave) and present a concrete protocol suite, CLIQUES, which offers complete key agreement services. The security analysis against active adversary is only heuristic. However, Pereira and Quisquater [74] have described a number of potential attacks, highlighting the need for ways to obtain greater assurance in the security of these protocols.

- Efficiency :

  Communication : For GDH.1, GDH.2, GDH.3, rounds required are respectively $2(n-1)$, n, $n+1$ and the respective messages sent per user are at most 2, 1, 2. For GDH.1, each of U_1, U_n sends only 1 message.

  Computation : Modular exponentiations computed by user U_i is $i+1$ for GDH.1 and $i+1$ for GDH.2. For GDH.3, modular exponentiations computed by user U_{n-1} are 2, user U_n are $n-1$ and for all other users are 4.

6.4 The Octopus Protocol and The Cube Protocol

(Becker, Willie [8], 1998)

In this work, Becker and Willie attempted to study lower bounds for the communication complexity of contributory key distribution. Their objective was to minimize the number of exchanges and to this aim, they introduced the basic octopus protocol without broadcasting which requires $2n-4$ exchanges. They have formally described the 2^d-cube protocol with d rounds and developed 2^d-octopus protocols with $\lceil \log_2 n \rceil + 1$ rounds that makes use of the basic octopus protocol. Both these protocols use no broadcasting.

- Protocol Description :

  Setup : Let G be a finite cyclic group of some large prime order q and g be a generator of G. We further assume a bijective mapping $\phi : G \rightarrow Z_q^*$.

  Key Agreement : Suppose the participants $U_1, \ldots, U_n$ want to agree on a common key. They perform the following steps.

(a) Octopus Protocol

Before introducing this protocol, first consider the following Diffie-Hellman key exchange among four user A, B, C, D. Users A and B and users C and D perform a Diffie-Hellman key exchange generating keys g^{ab} and g^{cd} respectively. Subsequently, $A(B)$ sends $g^{\phi(g^{ab})}$ to $C(D)$ and $C(D)$ sends $g^{\phi(g^{cd})}$ to $A(B)$. Hence, A and C (B and D) can generate the joint key $g^{\phi(g^{ab})\phi(g^{cd})}$.

In the octopus protocol, the participants $U_1, \dots, U_n$ are partitioned into five groups. Four users U_{n-3}, U_{n-2}, U_{n-1} and U_n take charge of the central control. We denote these users by A, B, C, D respectively. The remaining users are distributed in four groups: $\{U_i | i \in I_A\}$, $\{U_i | i \in I_B\}$, $\{U_i | i \in I_C\}$ and $\{U_i | i \in I_D\}$ where I_A, I_B, I_C, I_D are possibly of equal size, pairwise disjoint and $I_A \cup I_B \cup I_C \cup I_D = \{1, \dots, n-4\}$. Now $U_1, \dots, U_n$ can generate a group key as follows.

1. Each user $X \in \{A, B, C, D\}$ generates a joint key k_i with user U_i for all $i \in I_X$.
2. The users A, B, C, D perform the four party key exchange described above using the respective secret value $a = K(I_A)$, $b = K(I_B)$, $c = K(I_C)$ and $d = K(I_D)$, where $K(J) := \Pi_{i \in J} \phi(k_i)$ for $J \subseteq \{1, \dots, n-4\}$. Thereafter, A, B, C, D hold the joint and later group key
$$K := g^{\phi(g^{K(I_A \cup I_B)}) \phi(g^{K(I_C \cup I_D)})}.$$
3. We describe this step only for user A. The users B, C, D act correspondingly. For all $j \in I_A$, A sends $g^{K(I_B \cup I_A \setminus \{j\})}$ and $g^{\phi(g^{(I_C \cup I_D)})}$ to U_j. Now U_j calculates
$$(g^{K(I_B \cup I_A \setminus \{j\})})^{\phi(k_j)} = g^{K(I_A \cup I_B)}$$
and then generates the group key
$$K = (g^{\phi(g^{(I_C \cup I_D)})})^{\phi(g^{(K(I_A \cup I_B))})}.$$

(b) 2^d-Cube Protocol

In the cube protocol for 2^d-participants, the 2^d participants are identified with the vectors in the d-dimensional vector space $GF(2)^d$ and a basis $\vec{b}_1, \dots, \vec{b}_d$ of $GF(2)^d$ is chosen. The protocol may be performed in d rounds as follows.

1. In the first round, every participant $\vec{v} \in GF(2)^d$ generates a random number $r_{\vec{v}}$ and performs a Diffie-Hellman key exchange with participant $\vec{v} + \vec{b}_1$ using the values $r_{\vec{v}}$ and $r_{\vec{v}+\vec{b}_1}$.
2. In the i-th round, every participant $\vec{v} \in GF(2)^d$ performs a Diffie-Hellman key exchange with participant $\vec{v} + \vec{b}_i$, where both parties use the value generated in round $i-1$ as the secret value for the key exchange.

In every round i, $1 \leq i \leq d$, the participants communicate on a maximum number of parallel edges of the d-dimensional cube in the direction $\vec{b}_i$, Thus every party is involved in exactly one Diffie-Hellman exchange per round. Furthermore, all the parties share a common key at the end of this protocol because the vectors $\vec{b}_1, \dots, \vec{b}_d$ form a basis of the vector space $GF(2)^d$.

(c) 2^d-Octopus Protocol

In the 2^d-octopus protocol, participants act as in the octopus protocol with the only difference that 2^d instead of four users are distinguished to take charge of the central control, whereas remaining $n - 2^d$ users are partitioned into 2^d groups, *i.e.* in steps 1 and 3 of the octopus protocol, 2^d participants manage communication with the rest and in step 2 thus 2^d participants perform the cube protocol for 2^d participants.

- Assumption :
  DDH problem is hard.

- **Security :**
  It is proved that 2^d-cube protocol is secure against passive adversary. A very similar security analysis applies to 2^d-octopus protocol.

- Efficiency :

  Communication : For octopus protocol, 2^d-cube protocol and 2^d-octopus protocol, rounds required are respectively $2\lceil\frac{n-4}{4}\rceil + 2$, d and $2\lceil\frac{n-2^d}{2^d}\rceil + d$, messages sent per user are respectively $3n - 4$, nd and $3(n - 2^d) + 2^d d$.

  Computation : Both bijection operation (ϕ) and modular exponentiation required per user are at most $3n - 4$ for octopus protocol, nd for 2^d-cube protocol and $3(n - 2^d) + 2^d d$ for 2^d-octopus protocol.

6.5 Boyd and Nieto's Group Key Agreement

(Boyd, Nieto [21], 2003)

Boyd and Nieto [21] proposed a single-round authenticated static group key agreement that meets the bound of Becker and Wille [8] for a single-round protocol and have proved the security in the random oracle model following the model established by Bellare *et al.* [9, 12, 13]. The protocol does not provide forward secrecy.

- Protocol Description :

  Setup : Consider a secure public key encryption scheme $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where $\mathcal{K}$ is the key generation algorithm and $\mathcal{E}, \mathcal{D}$ are respectively the encryption and decryption algorithms. Let $\Sigma = (\overline{K}, \mathcal{S}, \mathcal{V})$ be a secure signature scheme with $\overline{\mathcal{K}}$ the key generation algorithm, $\mathcal{S}$ the signing algorithm and $\mathcal{V}$ the verification algorithm. The key distribution algorithm $\mathcal{G}_L$ assigns to each user U_i an encryption/decryption key pair $(e_i, d_i) \leftarrow K(1^k)$ and a signing/verification key pair $(\overline{e}_i, \overline{d}_i) \leftarrow \overline{K}(1^k)$ where k is a security parameter. Each user is provided by $\mathcal{G}_L$ an authenticated copy of the public keys of all other user. We also consider a one way hash function $h : \{0, 1\} \rightarrow \{0, 1\}^k$.

  Key Agreement : Let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be a set of n users wishing to establish a session key. The group members $U_1, \ldots, U_n$ consists of one distinguishing member, say, U_1, called initiator and all the other members are called responders. The users perform the following steps in order to agree upon a common key.

1. Each user U_i, chooses a random nonce $N_i \in \{0,1\}^k$.
2. Each responder U_i, $2 \leq i \leq n$, broadcasts $U_i|N_i$ to the rest of the users.
3. The initiator U_1 encrypts N_1 for each other user U_i in $\mathcal{U}$ using public encryption key e_i and generates $\mathcal{E}_{e_i}(N_1)$ for $2 \leq i \leq n$. Then U_1 signs $\mathcal{U}|\mathcal{E}_{e_2}(N_1)|\mathcal{E}_{e_3}(N_1)|\cdots|\mathcal{E}_{e_n}(N_1)$ to compute signature $\mathcal{S}_{d_1}(\mathcal{U}|\mathcal{E}_{e_2}(N_1)|\mathcal{E}_{e_3}(N_1)|\cdots|\mathcal{E}_{e_n}(N_1))$. U_1 broadcasts $\mathcal{U}|\{\mathcal{E}_{e_i}(N_i)|2 \leq i \leq n\}|\mathcal{S}_{d_1}(\mathcal{U}|\mathcal{E}_{e_2}(N_1)|\mathcal{E}_{e_3}(N_1)|\cdots|\mathcal{E}_{e_n}(N_1))$.
4. Each user computes the conference key $K_{\mathcal{U}} = h(N_1|N_2|\cdots|N_n)$.

- Assumption :
  The public key encryption scheme and the signature scheme are secure.

- Security :
  The protocol is proven to be secure in the random oracle model following the security model of [9, 12, 13]. However, the protocol does not provide forward secrecy.

- Efficiency :

  Communication : Round required is 1, initiator's broadcast constitutes $n+1$ messages and responder's broadcast constitutes only 2 messages.

  Computation : Each responder performs only 1 signature verification, 1 decryption in a public key cryptosystem and 1 operation of one-way hash function. The initiator has a heavy burden caused by $(n-1)$ encryptions in a public key cryptosystem and 1 signature generation. The computational burden of U_1 can be reduced substantially by careful choice of public key cryptosystem.
  The computation required are substantially lower than in the proven secure generalized Diffie-Hellman protocol of Bresson *et al.* [24, 25, 27] which require for user U_i to compute $i+1$ exponentiation in addition to generating and verifying a signature.

6.6 Bresson and Catalano's Group Key Agreement

(Bresson, Catalano [22], 2004)

A constant round provably authenticated static group key agreement protocol is introduced by Bresson and Catalano [22] which is based on secret sharing techniques combined with the El-Gamal encryption scheme and uses asynchronous network. Their security analysis is in the standard model under DDH assumption.

- Protocol Description :

Setup : Let p, q be two primes such that $q|p-1$. Suppose $G\langle g\rangle$ is a subgroup of order q, $\mathcal{H}$ is a hash function modeled as a random oracle and sid be the current session identity. Let users $U_1, \ldots, U_n$ want to agree upon a common session key. Each user U_i, for $1 \leq i \leq n$, has a public key h_i and a private key x_i such that $h_i = g^{x_i} \bmod p$. We also consider a secure signature scheme $\Sigma = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ where $\mathcal{K}, \mathcal{S}, \mathcal{V}$ are respectively the key generation, signing and verification algorithm.

In a preprocessing stage, each user U_i runs the key generation algorithm $\mathcal{K}$ to obtain a couple of matching signing and verification key $(\mathsf{SK}_i, \mathsf{PK}_i)$.

Key Agreement :

1. In the first round, each user U_i chooses randomly $a_i \in G$, $r, b_{i,1}, \ldots, b_{i,n-1} \in Z_q$ and define $f_i(z) = r_i + b_{i,1}z + b_{i,2}z^2 + \cdots + b_{i,n-1}z^{n-1} \bmod q$. Now for each j, $1 \leq j \leq n$, $j \neq i$, U_i chooses $k \in Z_q$ and sets $C_{i,j} = (A_{i,j}, B_{i,j}) = (g^k \bmod p, h_j^k a_i \bmod p)$. User U_i sends U_j the value $C_{i,j}$, $f_i(j)$ and $\sigma_{i,j} = \mathcal{S}_{\mathsf{SK}_i}(C_{i,j}|f_i(j)|\mathsf{sid})$.
2. In the second round, each user U_i, on receiving all the values above, first checks the authentication (signature) of all received values. If check fails, the user aborts the protocol (U_i also aborts the protocol in case he receives less than $n-1$ tuple $(C_{j,i}, f_j(i), \sigma_{j,i})$). U_i then multiplies the received cipher texts. Let $A_i = \Pi_{j\neq i}A_{j,i} \bmod p$ and $B_i = a_i\ \Pi_{j\neq i}B_{j,i} \bmod p$. U_i decrypts the result to define the value $a_{(i)} = B_i/A_i^{x_i}$ and computes $f_i = f_i(i) + \Sigma_{j\neq i}f_j(i) \bmod q$ as his share of a $(n-1)$-degree polynomial $f(z)$ whose free term is indicated by r. User U_i sends to other users the value f_i and $w_i = \mathcal{S}_{\mathsf{SK}_i}(f_i|\mathsf{sid})$.
3. In the third round, the users interpolate $f(z)$ and retrieve r. User U_i defines its session seed as
$$\mathsf{sk}_{(i)} = a_{(i)}\ g^r \bmod p.$$
4. For confirmation, each user U_i computes $s_i = \mathcal{H}(\mathsf{sk}_{(i)}|\mathsf{sid})$ and broadcasts this value together with its signature $\gamma_i = \mathcal{S}_{\mathsf{SK}_i}(s_i|\mathsf{sid})$. If the n broad-casted values are all the same, set the final key as
$$\mathsf{sk} = \mathcal{H}(\mathsf{sk}_{(i)}).$$

- Assumption :
DDH problem is hard and the signature scheme is secure.

- Security :
The protocol achieves provable security in the standard model under the well known DDH assumption.

- Efficiency :

Communication : Rounds required is 2 (plus a confirmation additional round).

Computation : Each user performs more than $3n$ modular exponentiations, $3n$ modular multiplications, n signature generations and n signature verifications.
The protocol is inefficient from point of view of computation rate.

7 Multi Party Dynamic Key Agreement

7.1 Bresson, Chevassut and Pointcheval's Group Key Agreement

(Bresson, Chevassut, Pointcheval [25], 2001)

Bresson *et al.* [25] provided a formal treatment of the authenticated group Diffie-Hellman key exchange problem in a scenario in which the membership is dynamic rather than static.

- Protocol Description :

  Setup : We consider a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$ where l is a security parameter. The session key SK associated to the protocol is $\{0,1\}^l$ equipped with a uniform distribution. Let $G = \langle g \rangle$ be a finite cyclic group of a k-bit prime (large) order q. This group could be a prime subgroup of Z_p^* or it could be an (hyper)-elliptic curve based group. We view G as a multiplicative group.

  – *Key Agreement* : The authenticated dynamic group key agreement scheme consists of three protocols SETUP1, REMOVE1 and JOIN1. Suppose a multi-cast group of users $\mathcal{I} = \{U_1, \ldots, U_n\}$ wish to agree upon a common key. They are arranged in a ring. Each user saves the set of values he receives in the down-flow of SETUP1, REMOVE1 and JOIN1. These values are needed to execute REMOVE1 for a subsequent removal of a user from $\mathcal{I}$. Any user from $\mathcal{I}$ could be selected as a group controller U_{GC} trusted to initialize the dynamic operations. In this protocol, we consider the user with the highest index in $\mathcal{I}$ as the group controller, the flows of a user U are signed using its long-lived key LL_U, the names of the users are in the protocol flows, and the session key sk is

$$\mathsf{sk} = \mathcal{H}(\mathcal{I}|\mathsf{Fl}_{\mathsf{max}(\mathcal{I})}|g^{x_1 \ldots x_{\mathsf{max}(\mathcal{I})}})$$

  where $\mathsf{Fl}_{\mathsf{max}(\mathcal{I})}$ is the down-flow, session identities SIDS and partner identities PIDS are appropriately defined.

(a) Protocol SETUP1

The protocol consists of two stages: up-flow and down-flow. The multi-cast group $\mathcal{I}$ is set to $\mathcal{J}$. In the up-flow, the user U_i receives a set (Y, Z) of intermediate value with

$$Y = \cup_{0<m<i}\{Z^{\frac{1}{x_m}}\}$$

and Z, where $Z = g^{\Pi_{0<t<i} x_t}$. User U_i chooses at random a private value x_i, raises the value in Y to the power of x_i and then concatenates with Z to obtain his intermediate value

$$Y' = \cup_{0<m\leq i}\{Z'^{\frac{1}{x_m}}\},$$

where $Z' = Z^{x_i} = g^{\Pi_{0<t\leq i} x_t}$. User U_i then forwards the value (Y', Z') to the next user in the ring. The down-flow occurs when $U_{\mathsf{max}(\mathcal{I})}$ receives the last up-flow. At that point $U_{\mathsf{max}(\mathcal{I})}$ performs the same steps as a user in the up-flow, but broadcasts the set of intermediate value Y' only. In effect, the value Z' computed by $U_{\mathsf{max}(\mathcal{I})}$ will lead to the session key sk, since $Z' = g^{\Pi_{0<t\leq n} x_t}$. Users in $\mathcal{I}$ compute sk and accept.

(b) Protocol REMOVE1

Suppose a set of users $\mathcal{J} \subset \mathcal{I}$ want to leave the multi-cast group $\mathcal{I}$. The multi-cast group $\mathcal{I}$ is first set to be $\mathcal{I} \backslash \mathcal{J}$. This protocol consists of a down-flow only. The group controller U_{GC} (*i.e.* with the highest-index in $\mathcal{I} \setminus \mathcal{J}$) generates a random value x'_{GC} and removes from the saved previous broadcast the values destinated to the users in $\mathcal{J}$. U_{GC} then raises all the remaining values in which x_{GC} appeared to the power of $(x_{\mathsf{GC}}^{-1}\, x_{\mathsf{GC}})$ and broadcasts the rest (x_{GC} is U_{GC}'s previous secret value). Users in $\mathcal{I}$ compute session key sk and accept. Users in $\mathcal{J}$ erase any internal data, user U_{GC} erases x_{GC} and $x_{\mathsf{GC}^{-1}}$ while internally saving x'_{GC}.

(c) Protocol JOIN1

Suppose a set of new users $\mathcal{J}$ want to join the multi-cast group $\mathcal{I}$. The protocol consists of two stages: up-flow and down-flow. The group controller U_{GC} (*i.e.* user with highest-index in $\mathcal{I}$) generates a random value x'_{GC}, raises the value from the saved previous broadcast in which x_{GC} appears to the power of $(x_{\mathsf{GC}}^{-1}\, x'_{\mathsf{GC}})$ and obtains a set of values Y' (x_{GC} is U_{GC}'s previous secret exponent). U_{GC} also computes the value Z' by raising the last value in Y' to x'_{GC}. U_i then forwards the values (Y', Z') to the first joining user in $\mathcal{J}$. From that point, JOIN1 will work as the SETUP1 protocol. Upon receiving the broadcast flow, users in $\mathcal{I} \cup \mathcal{J}$ erase previous session keys, compute sk and accept. The multi-cast group $\mathcal{I}$ is then set to $\mathcal{I} \cup \mathcal{J}$.

- Assumption :
  Generalized DH (or Many DH) problem is hard and the signature scheme is secure.

- Security :
  The protocol is proven to be secure in the random oracle model. The authors precisely define a security model for dynamic authenticated group key agreement with "implicit" authentication as the fundamental goal and the entity authentication as well and provide a concrete security analysis of their scheme in this model.

- Efficiency : Suppose n is the number of group members, j and l are respectively the number of joining and leaving users, e, v, s are respectively the cost of modular exponentiation, signature verification and signature generation.

Communication : For protocols SETUP1, REMOVE1 and JOIN1, rounds required are respectively $n, 1, O(j)$ and maximum bits sent per user are respectively $n|q| + |\sigma|$, $(n-l)|q| + |\sigma|$ and $(n+j)|q| + |\sigma|$.

Computation : For protocols SETUP1, REMOVE1 and JOIN1, computations required per user are respectively $ne + s + v$, $(n-l)e + s$ and $(n+j)e + 2s + jv$.

Note : Bresson, Chevassut and Pointcheval presented the static authenticated protocol in [27] and considered the dynamic scenario in [25]. The security analysis of both protocols are in a security model formalized by themselves in the random oracle model under the CDH assumption. To incorporate major missing details (*e.g.* strong-corruption and concurrent sessions), they further refine the existing security model and proposed an authenticated dynamic group Diffie-Hellman key exchange and show that it is provably secure under the DDH assumption within this model. Their security result holds in the standard model instead of random oracle model.

7.2 Bresson, Chevassut, Essiari and Pointcheval's Group Key Agreement

(Bresson, Chevassut, Essiari, Pointcheval [26], 2003)

A very efficient provably secure group key agreement is introduced in dynamic scenario which is suitable for restricted power devices and wireless environments. The scheme consists of three protocols: the setup protocol GKE.Setup, the remove protocol GKE.Remove and the join protocol GKE.Join. The main GKE.Setup protocol allows a cluster of mobile users (also called clients) and a wireless gateway (also called server) to agree on a session key. The other two protocols of the scheme handle efficiently the dynamic membership changes of clients in one wireless domain.

- Protocol Description :

  Setup : Let G be a finite multiplicative group of some large prime order q and g be a generator of G. Let $l = |q|$. We consider three hash functions $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$, $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^{l_0}$, $\mathcal{H}_1 : \{0,1\}^{l_1} \times G \rightarrow \{0,1\}^{l_0}$, where l_1 is the maximum bit length of a counter c used to prevent replay attacks. Let C be the set of all potential clients and S be the server. Before the protocol is run for the first time, an initialization phase occurs during which the following steps are performed.

  1. Each client $U_i \in C$ generates a pair of signing private/public keys $(\mathsf{SK}_i, \mathsf{PK}_i)$ by running the key generation algorithm of a signature scheme.
  2. The server S sets its private/public keys to be $(\mathsf{SK}_S, \mathsf{PK}_S) = (x, y)$, where $x \in_R Z_q^*$ and $y = g^x$.

  Key Agreement : The protocol executes in two rounds. In the first round, S collects contributions from individual clients and then, in the second round, it sends the group keying material to the clients.

(a) Protocol GKE.Setup

The algorithm, on input a set of client devices $\mathcal{J}$, performs the following steps.

1. Set the wireless group $\mathcal{G}_c$ to be the input set $\mathcal{J}$.
2. Each client $U_i \in \mathcal{G}_c$ chooses $x_i \in Z_q$ at random and precomputes $y_i = g^{x_i}$, $\alpha_i = \mathsf{PK}_S^{x_i} = y_i^{x_i}$ as well as a signature σ_i of y_i, under the private key SK_i.
3. Each client U_i sends (y_i, σ_i) to S.
4. For each $U_i \in \mathcal{G}_c$, the server S checks the signature σ_i using PK_i, and if they are all correct, computes the value $\alpha_i = y_i^x$.
5. The server S initializes the counter $c = 0$ as a bit-string of length l_1 and computes the shared secret value $K = \mathcal{H}_0(c|\{\alpha_i\}_{i \in \mathcal{G}_c})$ and sends to each client U_i the value c and $K_i = K \oplus \mathcal{H}_1(c|\alpha_i)$.
6. Each client U_i (and S) recovers the shared secret value $K = K_i \oplus \mathcal{H}_1(c|\alpha_i)$ and the session key $\mathsf{sk} = \mathcal{H}(K|\mathcal{G}_c|S)$.

(b) Protocol GKE.Remove

The algorithm on input the set $\mathcal{J}$ of leaving client devices, performs the following steps.

1. Update the wireless client group $\mathcal{G}_c = \mathcal{G}_c \setminus \mathcal{J}$.
2. The server S operates as in the GKE.Setup phase. It increases the counter c and computes the shared secret value $K = \mathcal{H}_0(c|\{\alpha_i\}_{i \in \mathcal{G}_c})$. item3. Then S sends to each client $U_i \in \mathcal{G}_c$ the value c and $K_i = K \oplus \mathcal{H}_1(c|\alpha_i)$.
4. Each client $U_i \in \mathcal{G}_c$ already holds the value $\alpha_i = g^{xx_i}$, and the old counter value. So it first checks that the new counter is greater than the old one and recovers the secret shared value $K = K_i \oplus \mathcal{H}_1(c|\alpha_i)$ and the session key $\mathsf{sk} = \mathcal{H}(K|\mathcal{G}_c|S)$.

(c) Protocol GKE.Join

The algorithm, on input the set of joining client devices $\mathcal{J}$, performs the following steps.

1. Update the wireless client group $\mathcal{G}_c = \mathcal{G}_c \cup \mathcal{J}$.
2. Each joining client $U_j \in \mathcal{J}$ chooses at random a value $x_j \in Z_q^*$ and precomputes $y_j = g^{x_j}$, $\alpha_j = \mathsf{PK}_S^{x_j}$ as well as a signature σ_j of y_j, under the private key SK_j
3. Each joining client $U_j \in \mathcal{J}$ sends the value (y_j, σ_j) to the device server S.
4. The server S checks the incoming signatures and if correct, operates as in the GKE.Setup phase, with an increased counter c and computes the shared value $K = \mathcal{H}_0(c|\{\alpha_i\}_{i \in \mathcal{G}_c})$.
5. Then it sends to each client $U_i \in \mathcal{G}_c$ the value c and $K_i = K \oplus \mathcal{H}(c|\alpha_i)$.
6. Each client $U_i \in \mathcal{G}_c$ already holds the value $\alpha_i = g^{xx_i}$ and the old counter value (set to be zero for the new ones). So it first checks the new counter is greater

than the old one, and recovers the secret shared value $K = K_i \oplus \mathcal{H}_1(c|\alpha_i)$ and the session key $\mathsf{sk} = \mathcal{H}(K|\mathcal{G}_c|S)$.

- Assumption :
  CDH problem is hard and the signature scheme is secure.

- Security :
  The protocol is proven to be secure in the random oracle model. They have formalized the adversarial model giving the adversary an enormous capabilities to closely model its abilities in the real life. They assume that the adversary never participates in the protocol execution (neither as a client, nor as a server). The security analysis of their scheme is in this security model and they claim that the protocol achieves partial forward secrecy. However, Nam, Kim, Won [71] demonstrate that the basic setup protocol GKE.Setup is insecure against known key security and if an active adversary is ever allowed to participate in the protocol as a client, the protocol does not even provide the fundamental security attribute implicit key authentication and perfect forward secrecy. They have proposed a modified version of the scheme which satisfies all the security properties: implicit key authentication, forward secrecy and known key security.

- Efficiency :

  Communication : Rounds required is 2 for each of GKE.Setup, GKE.Remove and GKE.Join.

  Computation : The protocol uses a base station as a trustee who has the special role of additing or removing clients from the group. This server computes n modular exponentiations, 1 signature generation, n signature verification, n one-way hash function operation and n XOR operations.

7.3 Nam, Kim, Kim and Won's Group Key Agreement

(Nam, Kim, Kim, Won [70], 2004)

Nam *et al.* [70] presented a communication-efficient dynamic group key agreement protocol well suited for a lossy and high-delay unbalanced network environment consisting of mobile hosts with restricted computational resources and stationary hosts with relatively high computational capabilities. They analyzed their scheme in the random oracle model and proved that it is secure under factoring assumption.

- Protocol Description :

  Setup : Let $N = pq$ where p, q are large distinct primes, $|p| = |q|$ such that $p = 2p' + 1$ and $q = 2q' + 1$ where p', q' are also prime integers. Then such an N is a Blum integer since $p = q = 3(\text{mod}4)$. We denote by Z_N^* the multiplicative group modulo N. We choose a quadratic residue $g \neq 1$ uniformly at random from the set of quadratic residues in Z_N^* generated by g. Suppose U_c is the controller in a multi-cast group $\mathcal{MG}$, and $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$ is a hash function modeled as a random oracle in the security proof of the scheme. In initialization phase, each user U_i is assigned a long-term public/private key pair $(\mathsf{PK}_i, \mathsf{SK}_i)$ by running a key generation algorithm. The public keys of all users are assumed to be known a priori to all parties including the adversary.

Key Agreement : The scheme consists of three algorithms IKA1, LP1 and JP1 for initial group formation, user leave and user join respectively.

(a) Protocol IKA1

Assume a multi-cast group $\mathcal{MG} = \{U_1, \ldots, U_n\}$ of n users wish to establish a session key by participating in initial group formation protocol IKA1 which runs in 2 rounds, one with $n-1$ unicast and the other with a single broadcast. They perform the following steps.

1. Each user U_i picks a random $r_i \in [1, N]$ and computes $z_i = g^{r_i} \bmod N$. Then $U_i (\neq U_c)$ signs $U_i | z_i$ to obtain signature σ_i and sends $m_i = U_i | z_i | \sigma_i$ to the controller U_c.
2. The controller U_c, on receiving each message m_i, verifies the correctness of m_i and computes $y_i = z_i^{r_c} \bmod N$. After receiving all the $n-1$ messages, U_c computes Y as

$$Y = \begin{cases} \Pi_{i \in [1,n] \setminus \{c\}} y_i \bmod N & \text{if } n \text{ is even} \\ \Pi_{i \in [1,n]} y_i \bmod N & \text{if } n \text{ is odd} \end{cases}$$

   U_c also computes the set $\mathcal{T} = \{T_i | i \in [1,n]\}$ where $T_i = Y \ y_i^{-1}$. Let $\mathcal{Z} = \{z_i | i \in [1,n]\}$. Then, U_c signs $\mathcal{MG}|\mathcal{Z}|\mathcal{T}$ to obtain signature σ_i and broadcasts $m_c = \mathcal{MG}|\mathcal{Z}|\mathcal{T}|\sigma_i$ to the entire group.
3. Each user $U_i (\neq U_c)$, on receiving the broadcast message m_c, verifies the correctness of m_c and computes $Y = z_c^{r_i} T_i \bmod N$. All users in $\mathcal{MG}$ compute their session key as $K = \mathcal{H}(\mathcal{T}|Y)$, and store their random exponent r_i and the set $\mathcal{Z}$ for future use.

(b) Protocol LP1

Consider a scenario where a set of user $\mathcal{L}$ leaves a multi-cast group $\mathcal{MG}_p$. Then protocol LP1 is executed to provide each user a new multi-cast group $\mathcal{MG}_n = \mathcal{MG}_p \setminus \mathcal{L}$ with a new session key. Any remaining user can act as the controller in the new multi-cast group $\mathcal{MG}_n$. The protocol LP1 requires only one round with a single broadcast and proceeds as follows.

1. The group controller U_c picks a new random $r_c \in [1, N]$ and computes $z'_c = g^{r'_c} \bmod N$. Using r'_c, z'_c and the saved set $\mathcal{Z}$, U_c then proceeds exactly as in

IKA1, except that it broadcasts $m_c = \mathcal{MG}|z_c|z'_c|\mathcal{T}|\sigma_c$ where z'_c is the random exponential from the previous controller.

2. Each user $U_i(\neq U_c)$, on receiving the broadcast message m_c, verifies that $\mathcal{V}(\mathcal{MG}_n|z_c|z'_c|\mathcal{T}, \sigma_c, \mathsf{PK}_c) = 1$ and the received z_c is equal to the one that is received in the previous session. All users in $\mathcal{MG}_n$ then compute their session key as $K = \mathcal{H}(\mathcal{T}|Y)$ and update the set $\mathcal{Z}$.

(c) Protocol JP1

Assume a scenario in which a set of j new users, $\mathcal{J}$, joins a multi-cast group $\mathcal{MG}_p$ to form a new multi-cast group $\mathcal{MG}_n = \mathcal{MG}_p \cup \mathcal{J}$. Then the user join protocol JP1 is executed to provide the users of $\mathcal{MG}_n$ with a new session key. The group controller in the new multi-cast group $\mathcal{MG}_n$ can be played by any user from the previous multi-cast group $\mathcal{MG}_p$. The protocol JP1 takes two rounds, one with j unicasts and the other with a single broadcast and proceeds as follows.

1. Each user $U_i \in \mathcal{J}$ picks a random $r_i \in [1, N]$ and computes $z_i = g^{r_i} \bmod N$. U_i then generates signature σ_i on $U_i|z_i$, sends $m_i = U_i|z_i|\sigma_i$ to U_c and stores r_i.
2. The group controller U_c proceeds in the usual way choosing a new random $r'_c \in [1, N]$, computes z'_c, Y, $\mathcal{T}$ and $K = \mathcal{H}(\mathcal{T}|Y)$, updating the set $\mathcal{Z}$ with new z_i's and then broadcasting $m_c = \mathcal{MG}_n|z_c|\mathcal{Z}|\mathcal{T}|\sigma_i$.
3. Each user $U_i \in \mathcal{MG}_p \setminus \{U_c\}$ verifies that the received z_c is equal to the one received in the previous session and user $U_i \neq U_c$ verifies the correctness of m_c. Then $U_i \neq U_c$ proceeds as usual, computing $Y = z_c'^{r_i} T_i \bmod N$ and $K = \mathcal{H}(\mathcal{T}|Y)$. All users in $\mathcal{MG}_n$ store or update the set $\mathcal{Z}$.

- Assumption :
  The factoring problem is hard and the signature scheme is existentially unforgeable.

- Security :
  A rigorous proof of security of the IKA1 protocol is provided in the security model as formalized by Bresson *et al.* [25, 24, 27]. The protocol is proven to be secure in the random oracle model under factoring assumption and provides perfect forward secrecy.

- Efficiency :

  Communication : For IKA1, JP1 and LP1 protocols, rounds required are respectively 2, 2, 1 and total messages sent are respectively n, $j + 1$ and 1 where j is the number of joining users. IKA1 requires $n - 1$ unicasts and 1 broadcast, JP1 requires j unicasts and 1 broadcast and LP1 requires 1 broadcast.

  Computation : For each of IKA1, JP1 and LP1 protocols, total modular exponentiation computed is $O(n)$ and total signature verification is $O(n)$.

Note : Nam, Kim, Yang, Won [72] investigate the problem of contributory group key agreement over combined wired/wireless networks, consisting of arbitrary number of mobile devices with limited computational resources and general-purpose stationary high-performance computer. They have designed a 3-round generalized protocol which take advantage of the difference in computing power among users and uses a 2-round unauthenticated protocol (introduced by them) very similar to the protocol IKA1 as a basic building block. Their 2-round basic protocol is proven to be secure against passive adversary under DDH assumption.

7.4 Kim, Lee and Lee's Group Key Agreement

(Kim, Lee, Lee [58], 2004)

A very efficient authenticated group key exchange (AGKE) scheme is developed by Kim *et al.* [58] for dynamically changing groups in *ad hoc* networks, where a member of a group may join and/or leave at any given time and a group key is exchanged without the help of any central server. Their proposed scheme is provably secure and requires constant rounds.

- Protocol Description :

  Setup : The protocol is based on the CDH assumption and a secure signature scheme $\Sigma = (\mathcal{K}, \mathcal{S}, \mathcal{V})$. We take the key space to be $\{0,1\}^l$ where l is a security parameter. Let G be a multiplicative group of some large prime order q $(l \leq |q|)$ and g be a generator of G. We also consider a one-way hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$. In an initialization phase, each user U_i is provided a public/private key pair $(\mathsf{PK}_{U_i}, \mathsf{SK}_{U_i})$ for verifying/signing. The list of public keys is published to all users.

  Key Agreement : The protocol consists of three algorithms GKE.Setup, GKE.Join and GKE.Leave for initial setup, user join and user leave respectively.

  (a) Protocol GKE.Setup

  Let $G_0 = \{U_1, \ldots, U_n\}$ be an initial group and ID_{U_i} be the identity of user U_i. We consider a ring structure among the members of G_0 and let $I_0 = \mathsf{ID}_{U_1}|\mathsf{ID}_{U_2}|\ldots|\mathsf{ID}_{U_n}$. The protocol executes in two rounds as follows.

  1. In round 1, each user U_i randomly chooses $k_i\{0,1\}^l$ and $x_i \in Z_q^*$, computes $y_i = g^{x_i}$ and keeps k_i secret. The last user U_n computes $\mathcal{H}(k_n|0)$. Each user U_i generates a signature $\sigma_i^{(1)}$ $\mathcal{S}_{\mathsf{SK}_{U_i}}(M_i^{(1)}|I_0|0)$ where $M_i^{(1)} = y_i$ for $1 \leq i \leq n-1$ and $M_n^{(1)} = \mathcal{H}(k_n|0)|y_n$, and broadcasts $M_i^{(1)}|\sigma_i^{(1)}$.
  2. In round 2, each user U_i, on receiving $M_j^{(1)}|\sigma_j^{(1)}$, verifies $\sigma_j^{(1)}$ on $M_j^{(1)}$ for all $j \neq i$. If some signatures are not valid, the process fails and halts. Otherwise, user U_i computes $t_i^L = \mathcal{H}(y_{i-1}^{x_i}|I_0|0)$, $t_i^R = \mathcal{H}(y_{i+1}^{x_i}|I_0|0)$ and generates $T_i = t_i^L \oplus t_i^R$. The last user U_n additionally computes $\widehat{T} = k_n \oplus t_n^R$. Each user U_i now generates signature $\sigma_i^{(2)} = \mathcal{S}_{\mathsf{SK}_{U_i}}(M_i^{(2)}|I_o|0)$ and broadcasts $M_i^{(2)} = \widehat{T}|T_n$.

3. In the key computation phase, each user U_i verifies signature $\sigma_j^{(2)}$, $j \neq i$. If all signatures are valid, U_i computes $\widetilde{t}_{i+1}^R, \widetilde{t}_{i+2}^R, \ldots, \widetilde{t}_{i+(n-1)}(=\widetilde{t}_i^L)$ by using t_i^R as follows.

$$\widetilde{t}_{i+1}^R = T_{i+1} \oplus t_i^R, \widetilde{t}_{i+2}^R = T_{i+2} \oplus \widetilde{t}_{i+1}^R, \ldots, \widetilde{t}_{i+(n-1)}^R = T_{i+(n-1)} \oplus \widetilde{t}_{i+(n-2)}^R.$$

Then U_i checks if $t_i^L = \widetilde{t}_i^L$ holds. Note that the above check enables honest users to notice the errors caused by system faults or wrong messages broadcatsed by illegal users and halts the protocol. However, it is not easy to find who transmitted illegal messages. Finally, each user obtains $\widetilde{k}_n$ from $\widehat{T}$ and checks if $\mathcal{H}(\widetilde{k}_n|0) = \mathcal{H}(k_n|0)$. This check value guarantees key control. All users then compute a session key $\mathsf{sk}_0 = \mathcal{H}(k_1|k_2|\ldots|k_{n-1}|k_n|0)$.

4. Each user U_i additionally computes $h_i^L = \mathcal{H}(y_{i-1}^{x_i}|\mathsf{sk}_0|0)$, $h_i^R = \mathcal{H}(y_{i+1}^{x_i}|\mathsf{sk}_0|0)$ and $X = \mathcal{H}(k_n|\mathsf{sk}_0|0)$, saves $(h_i^L, h_i^R, X, \mathsf{sk}_0)$ secretly and erases other ephemeral data. These post computed values will be used for subsequent join or leave operation.

(b) Protocol GKE.Join

Let $G_{v-1} = \{U_1, \ldots, U_n\}$ $(v \geq 1)$ be the current group and $\mathcal{J} = \{U_{n+1}, \ldots, U_{n+n'}\}$ $(n' \geq 1)$ be a set of new users wishing to join the group G_{v-1}. The group G_{v-1} is divided into three parts: $\{U_1\}$, $\{U_2, \ldots, U_{n-1}\}$ and $\{U_n\}$. We consider U_2 as a representative of $\{U_2, \ldots, U_{n-1}$ and for convenience of explanation, we allow that $U_{n+n'+1}$, $U_{n+n'+2}$ and $U_{n+n'+3}$ denote U_1, U_2, U_3 respectively. In this algorithm, a ring structure is considered among the users $U_{n+1}, \ldots, U_{n+n'+3}$. Let $\mathcal{G}$ be the set $\{U_{n+1}, \ldots, U_{n+n'+3}\}$ and $I_v = \mathsf{ID}_{U_1}|\ldots|\mathsf{ID}_{U_{n+n'}}$. The users in $\mathcal{G}$ proceed as follows.

1. In round 1, each user $U_{n+i}\mathcal{G}$ randomly chooses $k_{n+i} \in \{0,1\}^l$ and $x_{n+i} \in Z_q^*$, computes $y_{n+i} = g^{x_{n+i}}$ and keeps k_{n+i} secretly. The user $U_{n+n'+2}(=U_2)$ computes $y_{n+n'+2} = g^X$ by using the secret value X instead of $x_{n+n'+2}$ and the user $U_{n+n'+3}(=U_3)$ computes $\mathcal{H}(k_{n+n'+3}|v)$. Each user U_{n+i} generates signature $\sigma_{n+i}^{(1)} = \mathcal{S}_{\mathsf{SK}_{U_{n+i}}}(M_{n+i}^{(1)}|I_v|v)$ where $M_{n+i}^{(1)} = y_{n+i}$ for $1 \leq i \leq n'+2$ and $M_{n+n'+3}^{(1)} = \mathcal{H}(k_{n+n'+3}|v)|y_{n+n'+3}$ and broadcasts $M_{n+i}^{(1)}|\sigma_{n+i}^{(1)}$.
2. In round 2, all users, on receiving $M_{n+i}^{(1)}|\sigma_{n+i}^{(1)}$, verifies $\sigma_{n+i}^{(1)}$'s. Each user U_{n+i} computes $t_{n+i}^L = \mathcal{H}(y_{L(n+i)}^{x_{n+i}}|I_v|v)$, $t_{n+i}^R = \mathcal{H}(y_{R(n+i)}^{x_{n+i}}|I_v|v)$ and generates $T_{n+i} = t_{n+i}^L \oplus t_{n+i}^R$ where $L(n+i)$ and $R(n+i)$ respectively means the left and right index of $n+i$ on the ring for $i \in \{1, \ldots, n'+3\}$. The user $U_{n+n'3}$ additionally computes $\widehat{T} = k_{n+n'+3} \oplus t_{n+n'+3}^R$. Each user U_{n+i} generates signature $\sigma_{n+i}^{(2)} = \mathcal{S}_{\mathsf{SK}_{U_{n+i}}}(M_{n+i}^{(2)}|I_v|v)$ and broadcasts $M_{n+i}^{(2)}|\sigma_{n+i}^{(2)}$ where $M_{n+i}^{(2)} = k_{n+i}|T_{n+i}$ for $1 \leq i \leq n'+2\}$ and $M_{n+n'+3}^{(2)} = \widehat{T}|T_{n+n'+3}$. All users of $\{U_3, \ldots, U_{n-1}\}$ compute $t_{n+n'+2}^L$ and $t_{n+n'+2}^R$ by using X.

3. In the key computation phase, all users verify the signatures $\sigma_{n+i}^{(2)}$. If all signatures are valid, each user U_{n+i} computes $\widetilde{t}_{n+i+1}^{R}, \ldots, \widetilde{t}_{n+i+n'-1}(= \widetilde{t}_{n+i}^{L})$ by using t_{n+i}^{R} and checks if $t_{n+i}^{L} = \widetilde{t}_{n+i}^{L}$ holds. Also, users $U_3, \ldots, U_{n-1}$ can check it by using $t_{n+n'+2}^{L}$ and $t_{n+n'+2}^{R}$. Finally, all users recover $k_{n+n'+3}$ for $\widehat{T}$ and computes a new session key $\mathsf{sk}_v = \mathcal{H}(k_{n+1}| \ldots |k_{n+n'+3}|v)$.
4. Each new user U_{n+i} $(1 \leq i \leq n')$ computes $h_{n+i}^{L} = \mathcal{H}(y_{L(n+i)}^{x_{n+i}}|\mathsf{sk}_v|v)$ and $h_{n+i}^{R} = \mathcal{H}(y_{R(n+i)}^{x_{n+i}}|\mathsf{sk}_v|v)$. Users U_1 and U_n respectively compute $h_1^L = \mathcal{H}(y_{n+n'}^{x_1}|\mathsf{sk}_v|v)$ and $h_n^R = \mathcal{H}(y_{n+1}^{x_n}|\mathsf{sk}_v|v)$ instead of the previous value $h_i^L(= h_n^R)$. All users compute a new value $X = \mathcal{H}(k_n|\mathsf{sk}_v|v)$. Each user U_i then saves h_i^L, h_i^R, X and sk_v secretly and erases all other ephemeral data. These post computation is required for any subsequent join or leave operation.

(c) Protocol GKE.Leave

Let $G_{v-1} = \{U_1, \ldots, U_n\}$ be the current group and let $\mathcal{R} = \{U_{l_1}, \ldots, U_{l_{n''}}\}$ where $\{l_1, \ldots, l_{n''}\} \subset \{1, \ldots, n\}$, be a set of leaving users. We denote by $\mathcal{N}(\mathcal{R})$ the set of all left/right neighbors of leaving users on the ring. *i.e.*

$$\mathcal{N}(\mathcal{R}) = \{U_{L(l_1)}, U_{R(l_1)}, \ldots, U_{L(l_{n''})}, U_{R(l_{n''})}\}.$$

For the ease of discussion, let $U_{L(l_i)} = U_{l_i-1}$ and $U_{R(l_i)} = U_{l_i+1}$. Then $\mathcal{N}(\mathcal{R}) = \{U_{l_1-1}, U_{l_1+1}, \ldots, U_{l_{n''}-1}, U_{l_{n''}+1}\}$. To generate a new group $G_v = G_{v-1}\mathcal{R}$ with a new session key sk_v, a new Diffie-Hellman value should be shared between the left and right neighbors U_{l_j-1} and U_{l_j+1} $(1 \leq j \leq n'')$ respectively of the leaving user U_{l_j} on the ring. In this Leave algorithm, we consider a ring structure among members of G_v and we newly index the members as $G_v = \{U_1, \ldots, U_{n-n''}\}$. Let $I_v = \mathsf{ID}_{n_1}| \ldots |\mathsf{ID}_{n-n''}$. The algorithm runs in two rounds as follows.

1. In round 1, each user U_w of $\mathcal{N}(\mathcal{R})$ randomly chooses $k_w \in \{0,1\}^l$ and $x_w \in Z_q^*$, computes $y_w = g^{x_w}$ and keeps k_w secretly. Then user $U_{l_{n''}+1}$ computes $\mathcal{H}(k_{l_{n''}+1}|v)$. User U_w generates signature $\sigma_w^{(1)} = \mathcal{S}_{\mathsf{SK}_{U_w}}(M_w^{(1)}|I_v|v)$ where $M_w^{(1)} = y_w$ with $w \in \{l_1 - 1, l_1 + 1, \ldots, l_{n''} - 1\}$ and $M_{l_{n''}+1}^{(1)} = \mathcal{H}(k_{l_{n''}+1}|v)|y_{l_{n''}+1}$ and broadcasts $M_w^{(1)}|\sigma_w^{(1)}$.
2. All users of G_v verify signatures $\sigma_w^{(1)}$'s in the second round. If all signatures are valid, each user U_{l_j-1} (respectively U_{l_j+1}) of $\mathcal{N}(\mathcal{R})$ regenerates $h_{l_j-1}^{R} = y_{l_j+1}^{x_{l_j-1}}$ (respectively, $h_{l_j+1}^{L} = y_{l_j-1}^{x_{l_j+1}}$). Then each user U_i of G_v computes $t_i^L = \mathcal{H}(h_i^L|I_v|v)$, $t_i^R = \mathcal{H}(h_i^R|I_v|v)$ and $T_i = t_i^L \oplus t_i^R$. The user $U_{l_{n''}+1}$ generates a signature $\widehat{T} = k_{l_{n''}+1} \oplus t_{l_{n''}+1}^{R}$. Each user $U_i^{l_{n''}+1}$ generates a signature $\sigma_i^{(2)} = \mathcal{S}_{\mathsf{SK}_{U_i}}(M_i^{(2)}|I_v|v)$ and broadcasts $M_i^{(2)}|\sigma_i^{(2)}$ where $M_{l_{n''}+1}^{(2)} = \widehat{T}|T_{l_{n''}+1}$, $M_i^{(2)} = k_i|T_i$ for other users except $U_{l_{n''}+1}$ of $\mathcal{N}(\mathcal{R})$ and $M_i^{(2)} = T_i$ for users of $G_v \backslash \mathcal{N}(\mathcal{R})$.

3. In the session key computation phase, all users verify signatures $\sigma_i^{(2)}$'s. If all signatures are valid, each user U_i computes $\widetilde{t}_{i+1}^R, \widetilde{t}_{i+2}, \ldots, \widetilde{t}_{i+(n-n''-1)}(= \widetilde{t}_i^L)$ by using t_i^R and checks if $t_i^L = \widetilde{t}_i^L$ holds. Finally, all users compute a session key
$$\mathsf{sk}_v = \mathcal{H}(k_{l_1-1}|k_{l_1+1}|\ldots|k_{l_{n''}-1}|k_{l_{n''}+1}|v).$$
4. Each user U_i regenerates $h_i^L = \mathcal{H}(h_i^L|\mathsf{sk}_v|v)$, $h_i^R = \mathcal{H}(h_i^R|\mathsf{sk}_v|v)$ and $X = \mathcal{H}(k_{l_{n''}+1}|\mathsf{sk}_v|v)$ and saves h_i^L, h_i^R, X and the session key sk_v secretly. These post computation is required for any subsequent join or leave operation.

- Assumption :
  CDH problem is hard and the signature scheme is secure.
- Security :
  The protocol is proven to achieve provable security in the random oracle model. The proof technique is similar to that in [26].
- Efficiency :
  Suppose n is the number of group members, j and l are respectively the number of joining and leaving users, h, x, e, v, s are respectively the cost of hash function operation, XOR operation, modular exponentiation, signature verification and signature generation.

  Communication : For each of the algorithms GKE.Setup, GKE.Join and GKE.Leave, rounds required is 2, the maximum bits sent per user is $|q| + 3|h| + 2|\sigma|$ where $|q|, |h|, |\sigma|$ respectively denote the length of q, the order of the cyclic group G, the length of hash function h and the length of a signature.

  Computation : The computations required for algorithms GKE.Setup, GKE.Join and GKE.Leave are respectively $3e+4h+(n+1)x+2s+O(n)v$, $3e+4h+(j+1)x+2s+O(j)v$ and $3e + 4h + (n - 1)x + 2s + (l + n)v$.

8 Conclusion

This survey is devoted to realization of key agreement. We have included a comprehensive treatment of describing the most important key agreement protocols and provided an account of chronological developments connected with key agreement.

References

[1] S. Al-Riyami and K. G. Paterson. *Tripartite Authenticated Key Agreement Protocols from Pairings.* In proceedings of IMA Conference of Cryptography and Coding, LNCS 2898, pp. 332-359. Also available at http://eprint.iacr.org/2002/035.

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

[3] G. Ateniese, M. Steiner, and G. Tsudik. *Authenticated Group Key Agreement and Friends.* In proceedings of ACM CCS 1998[1], pp. 17-26, ACM Press, 1998.

[4] G. Ateniese, M. Steiner, and G. Tsudik. *New Multi-party Authenticated Services and Key Agreement Protocols.* In Journal of Selected Areas in Communications, 18(4), pp. 1-13, IEEE, 2000.

[5] P. S. L. M. Barreto. *Pairing Based Crypto Lounge.* Available at http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html

[6] P. S. L. M. Barreto, H. Y. Kim and M. Scott. *Efficient Algorithms for Pairing Based Cryptosystems.* In proceedings of Crypto 2002, LNCS 2442, pp. 354-368, Springer-Verlag, 2002. Also available at http://www.iacr.org/2002/008.

[7] R. Barua, R. Dutta, P. Sarkar. *Extending Joux Protocol to Multi Party Key Agreement.* In proceedings of Indocrypt 2003, LNCS 2904, pp. 205-217, Springer-Verlag, 2003. Also available at http://eprint.iacr.org/2003/062.

[8] K. Becker and U. Wille. *Communication Complexity of Group Key Distribution.* In proceedings of ACM CCS 1998, pp. 1-6, ACM Press, 1998.

[9] M. Bellare, R. Canetti, and H. Krawczyk. *A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols.* In proceedings of the 30th Annual Symposium on the Theory of Computing, pp. 419-428. ACM Press, 1998. Also available at http://www.cs.edu/users/mihir/papers/key-distribution.html/.

[10] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. *Relations Among Notions of Security for Public-key Encryption Schemes.* In proceedings of Crypto 1998, LNCS 1462, pp. 26-45, Springer-Verlag, 1998.

[11] M. Bellare, D. Pointcheval, and P. Rogaway. *Authenticated Key Exchange Secure Against Dictionary Attacks.* In proceedings of Eurocrypt 2000, LNCS 1807, pp. 139-155, Springer-Verlag, 2000.

[12] M. Bellare and P. Rogaway. *Entity Authentication and Key Distribution.* In proceedings of Crypto 1993, LNCS 773, pp. 231-249, Springer-Verlag, 1994.

[13] M. Bellare and P. Rogaway. *Provably Secure Session Key Distribution: The Three-party Case.* In proceedings of STOC 1995, pp. 57-66, ACM Press, 1995.

[14] M. Bellare and P. Rogaway. *Random Oracles are Practical : A Paradigm for Designing Efficient Protocols.* In proceedings of ACM CCS 1993, pp. 62-73, ACM Press, 1993.

[15] S. Blake-Wilson and A. Menezes. *Security Proofs for Entity Authentication and Authenticated Key Transport Protocols Employing Asymmetric Techniques.* In proceedings of the 5th International Workshop on Security Protocols, LNCS 1361, pp. 137-158, Springer-Verlag, 1997.

[16] S. Blake-Wilson, D. Johason and A. Menezes. *Key Agreement Protocols and Their Security Analysis.* In proceedings of the sixth IMA International Conference on Cryptography and Coding, LNCS 1355, pp. 30-45, Springer-Verlag, 1997.

[17] A. Boldyreva. *Efficient Threshold Signature, Multi-signature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-Group Signature Scheme.* In proceedings of PKC 2003, LNCS 2139, pp. 31-46, Springer-Verlag, 2003. Also available at http://www.iacr.org/2002/118.

[18] D. Boneh and M. Franklin. *Identity-Based Encryption from Weil Pairing.* In proceedings of Crypto 2001, LNCS 2139, pages 213-229, Springer-Verlag, 2001.

[19] D. Boneh, C. Gentry, B. Lynn and H. Shacham. *Aggregate and Verifiably Encrypted Signature from Bilinear Maps.* In proceedings of Eurocrypt 2003, LNCS 2248, pp. 514-532, Springer-Verlag, 2003.

[20] D. Boneh, B. Lynn, and H. Shacham. *Short Signature from Weil Pairing.* In proceedings of Asiacrypt 2001, LNCS 2248, pp. 213-229, Springer-Verlag, 2001.

[21] C. Boyd and J. M. G. Nieto. *Round-optimal Contributory Conference Key Agreement.* In proceedings of PKC 2003, LNCS 2567, pp. 161-174, Springer-Verlag, 2003.

[22] E. Bresson and D. Catalano. *Constant Round Authenticated Group Key Agreement via Distributed Computing.* In proceedings of PKC 2004, LNCS 2947, pp. 115-129, Springer-Verlag, 2004.

[23] E. Bresson, O. Chevassut and D. Pointcheval. *The Group Diffie-Hellman Problems.* In proceedings of SAC 2002, LNCS 2595, pp. 325-338, Springer-Verlag, 2002.

[24] E. Bresson, O. Chevassut, and D. Pointcheval. *Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions.* In proceedings of Eurocrypt 2002, LNCS 2332, pp. 321-336, Springer-Verlag, 2002.

[25] E. Bresson, O. Chevassut, and D. Pointcheval. *Provably Authenticated Group Diffie-Hellman Key Exchange - The Dynamic Case.* In proceedings of Asiacrypt 2001, LNCS 2248, pp. 290-309, Springer-Verlag, 2001.

[26] E. Bresson, O. Chevassut, A. Essiari and D. Pointcheval. *Mutual Authentication and Group Key Agreement for Low-power Mobile Devices.* Computer Communication, 27(17), pp. 1730-1737, 2004. A preliminary version appeared in proceedings of the 5th IFIP-TC6/IEEE , MWCN 2003, pp. 59-62, 2003. Full version available at http://www.di.ens.fr/ bresson.

[27] E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater. *Provably Authenticated Group Diffie-Hellman Key Exchange.* In proceedings of ACM CCS 2001, pp. 255-264, ACM Press, 2001.

[28] M. Burmester and Y. Desmedt. *A Secure and Efficient Conference Key Distribution System.* In proceedings of Eurocrypt 1994, LNCS 950, pp. 275-286, Springer-Verlag, 1995.

[29] M. Burmester and Y. Desmedt. *A Secure and Scalable Group Key Exchange System.* In Information Processing Letters, 94(3), pp. 137-143, 2005.

[30] R. Canetti and H. Krawczyk. *Analysis of Key-exchange Protocols and Their Use for Building Secure Channels.* In proceedings of Cryptographic Techniques, LNCS 2045, pp. 453-474. Springer-Verlag, 2001.

[31] Z. Chen. *Security Analysis on Nalla-Reddy's ID-based Tripartite Authenticated Key Agreement Protocol.* Available at http://eprint.iacr.org/2003/103.

[32] L. Chen and C. Kudla. *Identity Based Authenticated Key Agreement Protocols from Pairings.* Available at http://eprint.iacr.org/2002/184.

[33] K. Y. Choi, J. Y. Hwang and D. H. Lee. *Efficient ID-based Group Key Agreement with Bilinear Maps.* In proceedings of PKC 2004, LNCS 2947, Springer-Verlag, 2004.

[34] C. Cocks. *An Identity Based Encryption Scheme based on Quadratic Residues.* In Cryptography and Coding, LNCS 2260, pp. 360-363, Springer-Verlag, 2001. http://www.cesg.gov.uk/site/ast/idpkc/media/ciren.pdf.

[35] R. Cramer and V. Shoup. *A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Cipher text Attack.* In proceedings of Crypto 1998, LNCS 1462, pp. 13-25, Springer-Verlag, 1998.

[36] W. Diffie, M. Hellman. *New Directions in Cryptography.* In IEEE Transaction on Information Theory, IT-22 (6), pp. 644-654, 1976.

[37] R. Dutta, R. Barua and P. Sarkar. *Pairing Based Cryptographic Protocols : A Survey.* Manuscript 2004, submitted. Available at http://eprint.iacr.org/2004/064.

[38] R. Dutta, R. Barua and P. Sarkar. *Authenticated Multi-party Key Agreement : A Provably Secure Tree Based Scheme using Pairing.* In proceedings of National Workshop on Cryptology 2004, Kerala, India, October 2004.

[39] R. Dutta, R. Barua and P. Sarkar. *Provably Secure Authenticated Tree Based Group Key Agreement.* In proceedings of ICICS 2004, LNCS 3269, pp. 92-104, Springer-Verlag, 2004. Also available at http://eprint.iacr.org/2004/090.

[40] R. Dutta and R. Barua. *Dynamic Group Key Agreement in Tree-based Setting.* In proceedings of ACISP 2005, LNCS 3574, pp. 101-112, Springer-Verlag, 2005. Also available at http://eprint.iacr.org/2005/131.

[41] R. Dutta and R. Barua. *Constant Round Dynamic Group Key Agreement.* In proceedings of ISC 2005, LNCS, to appear on September 2005, Singapore. Also available at http://eprint.iacr.org/2005/221.

[42] R. Dutta and R. Barua. *Password-Based Encrypted Group Key Agreement.* Manuscript 2005, submitted.

[43] R. Dutta and R. Barua. *Group Key Agreement Immune to Dictionary Attacks.* Manuscript 2005, submitted.

[44] G. Frey, H. Ruck. *A Remark Concerning m-divisibility and The Discrete Logarithm in the Divisor Class Group of Curves.* In Mathematics of Computation, 62, pp. 865-874, 1994.

[45] S. Galbraith, K. Harrison and D. Soldera. *Implementing the Tate Pairing.* In proceedings of Algorithm Number Theory Symposium - ANTS V, LNCS 2369, pp. 324-337, Springer-Verlag, 2002.

[46] M. Girault and J. C. Paillers. *An Identity Based Scheme Providing Zero-knowledge Authenticated Key Exchange.* In proceedings of ESORICS 1990, pp. 173-184, 1990.

[47] F. Hess. *Efficient Identity Based Signature Schemes Based on Pairings.* In proceedings of SAC 2002, LNCS 2595, pp. 310-324, Springer-Verlag, 2002.

[48] I. Ingemarsson, D. T. Tang, and C. K. Wong. *A Conference Key Distribution System.* In IEEE Transactions on Information Theory 28(5), pp. 714-720, 1982.

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

[50] A. Joux. *A One Round Protocol for Tripartite Diffie-Hellman.* In proceedings of ANTS 4, LNCS 1838, pp. 385-394, Springer-Verlag, 2000.

[51] A. Joux, K. Nguyen. *Separating DDH from DH in Cryptographic Groups.* Available at http://eprint.iacr.org/2001/003.

[52] M. Just and S. Vaudenay. *Authenticated Multi-Party Key Agreement.* In proceedings of Asiacrypt 1996, LNCS 1163, pp. 36-49, Springer-Verlag, 1996.

[53] B. Kaliski. Contribution to ANSI X9F1 and IEEE P1363 working group, June 1998.

[54] J. Katz and M. Yung. *Scalable Protocols for Authenticated Group Key Exchange.* In proceedings of Crypto 2003, LNCS 2729, pp. 110-125, Springer-Verlag, 2003.

[55] Y. Kim, A. Perrig, and G. Tsudik. *Simple and Fault-tolerant Key Agreement for Dynamic Collaborative Groups.* In proceedings of ACM CCS 2000, pp. 235-244, ACM press, 2000.

[56] Y. Kim, A. Perrig, and G. Tsudik. *Tree Based Group Key Agreement.* Available at http://eprint.iacr.org/2002/009.

[57] Y. Kim, A. Perrig, and G. Tsudik. *Communication-efficient Group Key Agreement.* In proceedings of the 17th International Information Security Conference, IFIP SEC 2001, pp. 229-244, 2001.

[58] H. J. Kim, S. M. Lee and D. H. Lee. *Constant-Round Authenticated Group Key Exchange for Dynamic Groups.* In proceedings of Asiacrypt 2004, LNCS 3329, pp. 245-259, Sringer-Verlag, 2004.

[59] K. -K. R. Choo. *Revisit of McCullagh-Barreto Two-Party ID-Based Authenticated Key Agreement Protocols.* Available at http://eprint.iacr.org/204/343.

[60] 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. Also available at http://citeseer.nj.nec.com/law98efficient.

[61] H. K. Lee, H. S. Lee and Y. R. Lee. *Multi-party Authenticated Key Agreement Protocols from Multi linear Forms.* Available at http://eprint.iacr.org/2002/166.

[62] T. Matsumoto, Y. Takashima and H. Imai. *On Seeking Smart Public-key Distribution Systems.* In Transactions of the IECE of Japan, E69, pp. 99-106, 1986.

[63] 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. Also available at http://eprint.iacr.org/2004/122.

[64] A. Menezes, T. Okamoto, and S. Vanstone. *Reducing Elliptic Curve Logarithms to Logarithms in a finite field.* In IEEE Transaction on Information Theory, 39, pp. 1639-1646, 1993.

[65] A. Menezes, P. C. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, 1997. Also available at http://cacr.math.uwaterloo.ca/hac.

[66] D. Nalla and K. C. Reddy. *Identity Based Authenticated Group Key Agreement Protocol.* In proceedings of Indocrypt 2002, LNCS 2551, pp. 215-233, Springer-Verlag, 2002.

[67] D. Nalla. *ID-Based Tripartite Key Agreement with Signature.* Available at http://eprint.iacr.org/2003/144.

[68] D. Nalla and K. C. Reddy. *ID-Based Tripartite Authenticated Key Agreement Protocols from Pairings.* Available at http://eprint.iacr.org/2003/004.

[69] J. Nam, J. Lee, S. Kim and D. Won. *DDH-based Group Key Agreement for Mobile Computing.* Available at http://eprint.iacr.org/2004/127.

[70] J. Nam, S. Kim, S. Kim and D. Won. *Provably-Secure and Communication-Efficient Scheme for Dynamic Group Key Exchange.* Available at http://eprint.iacr.org/2004/115.

[71] J. Nam, S. Kim and D. Won. *Attacks on Bresson-Chevassut-Essiari-Pointcheval's Group Key Agreement Scheme for Low-Power Mobile Devices.* Available at http://eprint.iacr.org/2004/251.

[72] J. Nam, S. Kim, H. Yang and D. Won. *Secure Group Communications over Combined Wired/Wireless Network.* Available at http://eprint.iacr.org/2004/260.

[73] E. Okamoto. *Proposal for Identity Based Key Distribution System.* In Electronic Letters, 22, pp. 1283-1284, 1986.

[74] O. Pereira and J.J. Quisquater. *A Security Analysis of the Cliques Protocol Suite.* In Computer Security Foundations Workshop (CSFW 2001), pp. 73-81, IEEE Computer Society Press, 2001.

[75] M. Scott. *Authenticated ID-based Key Exchange and Remote Log-in with Insecure Token and PIN Number.* Available at http://eprint.iacr.org/2002/164.

[76] A. Shamir. *Identity-based Cryptosystems and Signature Schemes.* In proceedings of Crypto 1984, LNCS 196, pp. 47-53, Springer-Verlag, 1984.

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

[78] K. Shim. *Cryptanalysis of Al-Riyami-Paterson's Authenticated Three Party Key Agreement Protocols.* Available at http://eprint.iacr.org/2003/122.

[79] K. Shim. *Cryptanalysis of ID-Based Tripartite Authenticated Key Agreement Protocol.* Available at http://eprint.iacr.org/2003/115.

[80] V. Shoup. *On Formal Models for Secure Key Exchange.* In IBM Technical Report RZ 3120, 1999. Also available at http://shoup.net/papers.

[81] N. P. Smart. *An Identity-based Authenticated Key Agreement Protocol Based on the Weil Pairing.* In Electronic Letters, 38, pp. 630-632, 2002. Also available at http://www.iacr.org/2001/111.

[82] B. Song and K. Kim. *Two-pass Authenticated Key Agreement Protocol with Key Confirmation.* In proceedings of Indocrypt 2000, LNCS 1977, pp. 237-249, Springer-Verlag, 2000.

[83] M. Steiner, G. Tsudik, M. Waidner. *Diffie-Hellman Key Distribution Extended to Group Communication.* In proceedings of ACM CCS 1996, pp. 31-37, ACM Press, 1996.

[84] M. Steiner, G. Tsudik and M. Waidner. *Cliques : A New Approach to Group Key Agreement.* In IEEE Conference on Distributed Computing Systems, May 1998, pp. 380.

[85] H. M. Sun and B. T. Hsieh. *Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings.* Available at http://eprint.iacr.org/2003/113.

[86] G. Xie. *Cryptanalysis of Noel McCullagh and Paulo S. L. M. Barreto's Two-party Identity-Based Key Agreement.* Available at http://eprint.iacr.org/2004/308.

[87] G. Xie. *An ID-Based Key Agreement Scheme from Pairing.* Available at http://eprint.iacr.org/2005/093.

[88] F. Zhang, S. Liu and K. Kim. *ID-based One Round Authenticated Tripartite Key Agreement Protocol with Pairings.* Available at http://eprint.iacr.org/2002/122.