HGI Network and Data Security Group Technical Report 2006/03

Security-Focused Survey on Group Key Exchange Protocols

Mark Manulis

Horst-Görtz Institute Ruhr-University of Bochum

Security-Focused Survey on Group Key Exchange Protocols

Mark Manulis
Horst Görtz Institute for IT-Security
Ruhr University Bochum, Germany
mark.manulis@rub.de

07.11.2006

Abstract

In this paper we overview a large number of currently known group key exchange protocols while focusing on the protocols designed for more than three participants (for an overview of two- and three-party key exchange protocols we refer to [BM03, DB05c]). For each mentioned protocol we briefly describe the current state of security based on the original analysis as well as later results appeared in the literature. We distinguish between (i) protocols with heuristic security arguments based on informally defined security requirements and (ii) protocols that have been proven secure in one of the existing security models for group key exchange. Note, this paper continues the work started in [Man06] which provides an analytical survey on security requirements and currently known models for group key exchange. We emphasize that the following survey focuses on the security aspects of the protocols and does not aim to provide any efficiency comparison. The reader interested in this kind of surveys we refer to [RH03, AKNRT04].

Keywords: group key exchange, protocol security, survey, analysis

Contents

1 Introduction

Group key exchange (GKE) protocols provide participants with a shared secret key (group key) which can be further used to achieve confidentiality and authentication in different group applications. This paper considers various group key exchange protocols proposed in the literature whereby focusing on their security arguments concerning the resistance against different types of attacks. We distinguish between (i) protocols with heuristic security arguments based on informally defined security requirements and (ii) protocols that have been proven secure in one of the existing security models for group key exchange. This paper continues the work started in [Man06] which describes and analyzes various informally defined security requirements and currently known formal security models for group key exchange protocols.

In the following we briefly overview the main security requirements that are most relevant for the following analytical survey. The description is done along the lines in [Man06]. We also assume that the reader has basic knowledge of cryptography.

1.1 Short Overview of Security Requirements for Group Key Exchange Protocols

One of the basic security requirements for GKE protocols is *(authenticated) key exchange security* (AKE-security) [BCPQ01] that ensures indistinguishability of the computed session group key from a random number (also known as semantic security of the protocol) assuming that an active adversary who controls the underlying communication channel (eavesdrops, modifies and injects messages) is able to reveal group keys of other protocol sessions. Note that this requirement also deals with impersonation attacks where an adversary tries to impersonate participants of the protocol. AKE-security comes currently in two flavors w.r.t. the notion of *forward secrecy* that considers damages to the secrecy of session group keys resulting from adversarial actions in later sessions: (i) AKE-security with *weak forward secrecy* where the adversary is allowed to reveal long-lived keys of participants, and (ii) AKE-security with *strong forward secrecy* where the adversary is additionally allowed to reveal internal memory of participants, e.g., their ephemeral secrets. Attacks that reveal internal memory are also called *strong corruptions* [Sho99, Ste02, BCP02a]. Note that AKE-security considers adversaries who are not legitimate session participants since every legitimate participant learns the established session group key implicitly. Almost all currently available security models, i.e., BCPQ [BCPQ01], BCP [BCP01], BCP$^+$ [BCP02a], KY [KY03], KS/UC-KS [KS05], and BVS [BVS05], provide definitions concern-

ing AKE-security, using different versions of forward secrecy[1].

Another basic security requirement for GKE protocols according to the BCPQ and BCP models is *mutual authentication security* (MA-security) to ensure that all legitimate protocol participants and only them compute identical session group keys, thus this requirement also subsumes the property of key confirmation. As noticed in the KS/UC-KS model, these requirements must also hold in case that the adversary is represented by a (subset of) legitimate participant(s) whose behavior deviates from the protocol specification, i.e., *malicious participant(s)*. According to [CBH05] consideration of malicious participants is also of prime importance to prevent unknown-key share attacks where an active adversary tries to make one protocol participant believe that the group key is shared with one party when it is in fact shared with another party.

In [MWW98], Mitchel *et al.* described the notion of *key control*, i.e., an attack where an adversary tries to influence the resulting value of the computed session group key[2]. Note that opposite to group key distribution protocols, in group key exchange protocols no party should be able to choose the resulting group key on behalf of other participants. Ateniese *et al.* [AST98] proposed a related notion called *contributiveness* that encompasses the fact that all protocol participants must equally contribute to the computation of the group key. Two different versions of key control and contributiveness are can be currently found in the literature. A weaker form like implicitly considered by Bresson and Catalano in [BC04] assumes honest protocol participants that have biased source of randomness so that a curious adversary can possibly gain extra information and break the AKE-security of the protocol. A stronger version like considered by Bohli *et al.* in [BVS05] assumes malicious participants that try to influence honest participants computing some special value as the resulting group key. Note that according to [Man06] none of the currently available security models for GKE protocols provides definitions concerning key control and contributiveness that consider strong corruptions.

Our survey focuses on the security aspects (AKE-/MA-security, key control and contributiveness) of static and dynamic group key exchange protocols while *also* considering strong corruptions.

Many group key exchange protocols described in our survey can be seen as extensions of the following two well-known protocols: two-party key exchange by Diffie and Hellman [DH76] and three-party key exchange by Joux [Jou00].

[1] See [Man06] or original papers for more details concerning the mentioned security models.

[2] Although [MWW98] considers key control for two-party protocols similar threats become even more important in a group setting.

1.2 Two-Party Key Exchange Protocol by Diffie and Hellman

The protocol proposed by Diffie and Hellman in [DH76] is the earliest key exchange protocol that allows two participants, U_1 and U_2, compute a secret key k over a public communication channel. Mathematical operations of the protocol are performed in a multiplicative group $\mathbb{G}$ where the well-known Discrete Logarithm (DL) problem is believed to be intractable. Let g be a generator of $\mathbb{G}$. Figure 1 describes the generalized version of the protocol. Obviously, the resulting shared key has the form $k = g^{x_1 x_2}$. The seman-

- Each U_i, $i \in \{1, 2\}$ chooses a random $x_i \in_R \mathbb{Z}_q$ and sends $z_i := g^{x_i}$ to U_{3-i}.
- Each U_i, $i \in \{1, 2\}$ computes $k_i := (z_{3-i})^{x_i}$.

Figure 1: Two-Party Key Exchange Protocol by Diffie and Hellman [DH76]

tic security of k against passive adversaries relies on the Decisional Diffie-Hellman (DDH) assumption. The original Diffie-Hellman protocol does not provide protection against impersonation attacks. A large number of variations has been proposed after the invention of the protocol to improve its security degree, the most recent are [LMQ+03, Kra05]. Mostly all group key exchange protocols considered in our survey can be seen as more or less complex extensions related to this original Diffie-Hellman key exchange protocol.

1.3 Three-Party Key Exchange by Joux

In [Jou00], Joux proposed the following efficient key exchange protocol designed for three participants. The protocol uses a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ where $\mathbb{G}_1$ is an additive group of prime order q and $\mathbb{G}_2$ a multiplicative group of the same order, e.g., $\mathbb{G}_1$ is a subgroup of the group of points on an elliptic curve E over a finite field, $\mathbb{G}_2$ a subgroup of a multiplicative group over a related finite field, and $\hat{e}$ is an appropriate pairing on E (we refer to [BSS05] for more details on pairings in elliptic curves). Also, an element (point) $P \in \mathbb{G}_1$ with $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$ should be publicly known. The protocol between U_0, U_1, and U_2 proceeds as follows. Obviously, at the end of the protocol each user computes

- Each U_i chooses $x_i \in_R \mathbb{Z}_q^*$ and broadcasts $y_i := x_i P$ to all other users.
- Each U_i computes $k_i := \hat{e}(y_{(i+1) \bmod 3}, y_{(i+2) \bmod 3})^{x_i}$.

Figure 2: Three-Party Key Exchange Protocol by Joux [Jou00]

the group key $k = \hat{e}(P, P)^{x_0 x_1 x_2}$. The protocol requires only one communication round. Although not explicitly shown in [Jou00], the semantic security of the protocol against passive adversaries is based on the Bilinear Diffie-Hellman (BDH) assumption [BF03]. Joux'

protocol does not provide any form of authentication. Several attempts have been done to add authentication to the Joux' protocol, e.g., certification-based [ARP03, Shi03a, HBN04] and identity-based [ZLK02, NK03, Nal03, Shi03b, CVC04] protocol some of which could be broken in [Che03, Shi03c, Shi03b, SH03].

1.4 Relationship between Group Key Exchange Protocols

Regardless of the separation into group key exchange protocols with heuristic security arguments and protocols with security proofs in the available security models some of the protocols included in our survey have certain similarities which we describe in the following.

The protocols in Sections 3.1, 3.2, 3.3, and 3.4 can be considered as modifications of the static group key exchange protocol proposed by Burmester and Desmedt [BD94] which we describe in Section 2.1. These protocols are characterized by the constant number of communication rounds and are, therefore, scalable for large groups. Some of these protocols derive the group key from bases whose discrete logarithms are outputs of an additive cyclic function.

The protocols in Sections 2.3, 2.4, and 3.6 can be considered as modifications of the static group key exchange protocol proposed by Ingemarsson, Tang, and Wong [ITW82] which we describe in Section 2.2. Most of these protocols derive the group key from bases whose discrete logarithms are outputs of a symmetric multiplicative function. In particular, from the value of the form $g^{x_1 \cdots x_n}$ where g is a generator of a cyclic group $\mathbb{G}$ where the DL problem is believed to be intractable, and every x_i, $i \in [1, n]$ is a private exponent of participant U_i. This form can be seen as a "natural" extension of the two-party Diffie-Hellman key exchange protocol described above.

The protocols in Sections 2.6, 2.7, derive the group key from a value obtained by an iterative application of the two-party Diffie-Hellman protocol. The earliest protocol of this class was proposed by by Steer, Strawczynski, Diffie, and Wiener [SSDW90] which we describe in Section 2.5. Most of the protocols of this class arrange participants into a logical binary tree structure which is either linear or balanced. In general, each user is logically assigned to a leaf node of a binary tree T. We use *labels* $\langle l, v \rangle$ to uniquely identify a node of a tree where $l \in \{0, d_T\}$ is a corresponding level of T, d_T the depth of T, and $v \in \mathbb{N}$ the nodes' position within the level. Note that in linear binary trees the depth d_T is linear in the number of participants whereas in the balanced binary trees d_T is logarithmic. Figure 3 shows an example of both tree types for $n = 4$. Every node of the tree contains a pair $(x_{\langle l,v \rangle}, y_{\langle l,v \rangle})$ where $x_{\langle l,v \rangle}$ is considered to be a secret key, and $y_{\langle l,v \rangle}$ a public value derived from $x_{\langle l,v \rangle}$. The root of the tree, denoted $\langle 0, 0 \rangle$

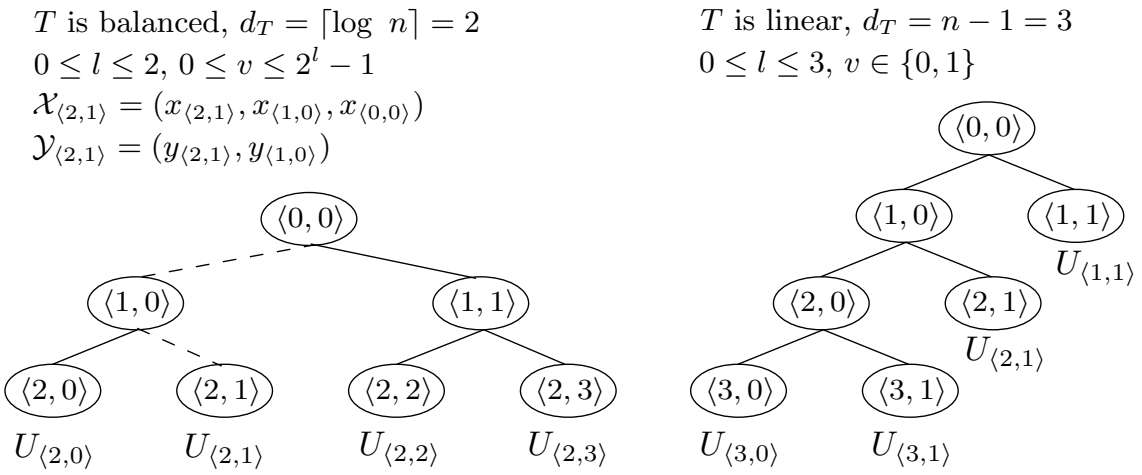

Figure 3: Example: Balanced and Linear Trees for $n = 4$

contains only the secret value $x_{\langle 0,0\rangle}$, which is usually used in the protocols to derive the resulting group key k. By a *secret key path of a node* $\langle l, v\rangle$ we denote the list $\mathcal{X}_{\langle l,v\rangle} := (x_{\langle l,v\rangle}, x_{\langle l-1,\lfloor v/2\rfloor\rangle}, \ldots, x_{\langle 1,\lfloor v/2^{l-1}\rfloor\rangle}, x_{\langle 0,0\rangle})$, and by a *public key path of a node* $\langle l, v\rangle$ the corresponding list $\mathcal{Y}_{\langle l,v\rangle} := (y_{\langle l,v\rangle}, y_{\langle l-1,\lfloor v/2\rfloor\rangle}, \ldots, y_{\langle 1,\lfloor v/2^{l-1}\rfloor\rangle})$. The protocols differ in a way of how this group key is computed. Furthermore, some of the described protocols update the logical tree structure and the the secret value $x_{\langle 0,0\rangle}$ upon occurring dynamic changes of the group formation.

The protocols in Sections 2.8, 2.9, and 3.7 arrange participants into a ternary tree like the one in Figure 4. These protocols are extensions of the Joux' three-party key exchange protocol from Section 1.3.

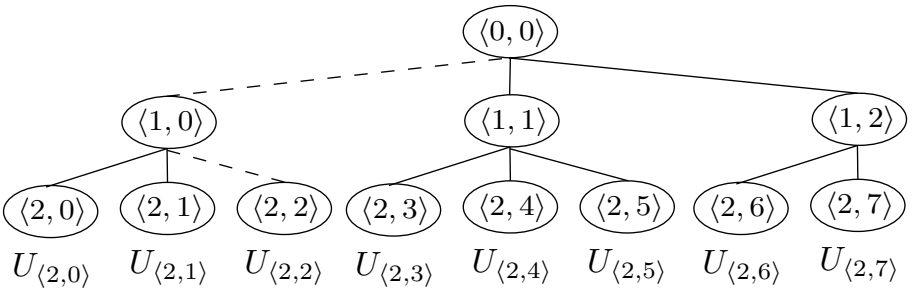

Figure 4: Example: Balanced Ternary Tree for $n = 8$

2 Group Key Exchange Protocols with Heuristic Security Arguments

2.1 Protocol by Burmester and Desmedt

Burmester and Desmedt [BD94, BD05] describe several protocols that allow a set of n users (group members) $U_1, \ldots, U_n$ to compute a secret group key k. The proposed protocols differ with respect to the underlying network topology. Although the majority of their protocols belongs to the class of group key distribution there are two protocols that can be considered as group key exchange protocols between users that are connected either over a broadcast network or other a bi-directional cyclic network. In the following we give a brief description of the static protocol designed for a broadcast network (denoted in [BD94] as Protocol 3).

All group members are logically ordered into a cycle, i.e., the indices are taken modulo n so that user U_0 is U_n and user U_{n+1} is U_1. All mathematical operations are performed in a cyclic group $\mathbb{G} \subseteq \mathbb{Z}_p$ with prime p generated by $g \in \mathbb{Z}_p$ of order q. It is assumed that the description of $\mathbb{G}$ is implicitly known to all users. The protocol proceeds as follows.

- Each U_i chooses a random $r_i \in_R \mathbb{Z}_q$ and broadcasts $z_i := g^{r_i} \pmod p$.
- Each U_i broadcasts $X_i := (z_{i+1}/z_{i-1})^{r_i} \pmod p$.
- Each U_i computes $k_i := (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i+n-2} \pmod p$ for $i = 1, \ldots, n$.

Figure 5: Protocol by Burmester and Desmedt [BD94]

Note that after the protocol is completed every user holds the same group key $k = k_i = g^{r_1 r_2 + r_2 r_3 + \ldots + r_n r_1} \pmod p$. The protocol designed for bi-directional cyclic networks (denoted in [BD94] as Protocol 4) proceeds similar except that corresponding messages are sent between any two directly connected participants. This increases the communication and computation overhead compared to the above protocol.

The heuristic security proof given in the pre-proceedings version considers only key secrecy requirement with respect to passive eavesdroppers under the well-known Computational Diffie-Hellman (CDH) assumption. The insecurity against active adversaries follows from the absence of authentication. The authors also mention a variant of an authenticated protocol where each user U_i authenticates corresponding z_i to the subsequent user U_{i+1} using a zero-knowledge proof technique [CEvdG87]. However this technique does not provide security against impersonation attacks due to the missing identification.

2.1.1 Variants by Choi *et al.* and Manulis

Choi *et al.* [CHL04] proposed a variant of the unauthenticated Burmester-Desmedt protocol based on the technique of bilinear pairings [BF03, BSS05]. The heuristic security analysis considers only indistinguishability of group keys from random numbers with respect to passive adversaries under the so-called Decisional Bilinear Diffie-Hellman (DBDH) assumption [Jou00, BF03, BSS05] in the Random Oracle Model (ROM) [BR93]. Choi *et al.* have also constructed a protocol that provides identity-based authentication. Its security is argued in ROM under the so-called Decisional Hash Bilinear Diffie-Hellman (DHBDH) assumption [BDS03], which is a non-standard cryptographic assumption strictly stronger than DBDH. Later, Zhang and Chen [ZC03] showed an attack against the authentication property of the identity-based version of Choi *et al.*'s protocol where two malicious participants impersonate an honest participant using his authentication transcript from some previous protocol execution.

Manulis [Man05] described an elliptic curve equivalent of the original Burmester-Desmedt protocol in the context of mobile ad-hoc communication. The deployment of the elliptic curve cryptography results in a better trade-off between computation and communication costs due to the smaller sizes of the operands. The informally argued semantic security relies on the elliptic-curve version of the Decisional Diffie-Hellman (ECDDH) assumption [BSS05] if a non-supersingular and a non-trace-2 elliptic curve is used as required in [JN03].

2.2 Protocol by Ingemarsson, Tang, and Wong

In [ITW82], Ingemarsson, Tang, and Wong proposed a family of group key exchange protocols from which we describe the mostly known one in Figure 6. It is assumed that all participants $U_1, \ldots, U_n$ implicitly know the description of the multiplicative group $\mathbb{G}$ of prime order q with the corresponding generator g. Participants are logically ordered into a cycle (similar to the Burmester-Desmedt protocol), i.e., the indices are taken modulo n so that member U_0 is U_n and member U_{n+1} is U_1. At the end of the protocol every U_i com-

- In round 0, each U_i chooses a random $x_i \in \mathbb{Z}_q^*$, computes g^{x_i} and forwards it to U_{i+1}.
- In round t, $t = 1, \ldots, n-2$ each U_i computes $g^{\prod\{x_j | j \in [i-t,i]\}}$ (using x_i as an exponent for $g^{\prod\{x_j | j \in [i-t,i-1]\}}$ received in the previous round) and forwards it to U_{i+1}.

Figure 6: Protocol by Ingemarsson, Tang, and Wong [ITW82]

putes the group key $k := g^{x_1 \cdots x_n}$. The informal security proof considers semantic security against passive adversaries under the DDH assumption. The absence of the authentication

implies the insecurity against active adversaries.

2.3 Protocols by Steiner, Tsudik, and Waidner

Steiner *et al.* [STW96] proposed a generic group key exchange protocol and three realizations, called GDH.1, GDH.2, and GDH.3, respectively. The generic construction considers a cyclic group $\mathbb{G}$ of prime order q generated by g. Through a distributed computation of the subsets of $\{g^{\prod X} | X \subset \{x_1, \ldots, x_n\}\}$ every member U_i computes $g^{x_1 \ldots x_{i-1} x_{i+1} \ldots x_n}$. This allows every U_i to derive the resulting secret group key k_i with an additional exponentiation of the mentioned value with the private exponent x_i. The first proposed realization, i.e., the protocol GDH.1 consists of the two stages: upflow and downflow, described in Figure 7. The second protocol GDH.2 consists of the upflow stage and an additional broadcast round, and proceeds as described in Figure 8. The third protocol GDH.3 consists of the upflow stage, two broadcast rounds and one response round and proceeds as described in Figure 9.

- *Upflow:* In round i, $i = 1, \ldots, n-1$ the user U_i chooses a random $x_i \in \mathbb{Z}_q^*$ and forwards $\{g^{\prod\{x_t | t \in [1,j]\}} | j = 1, \ldots, i\}$ to U_{i+1}.
- *Downflow:* In round $n-1+i$, $i = 1, \ldots, n-1$ the user U_{n-i+1} forwards $\{g^{\prod\{x_t | t \in [1,n] \wedge t \notin [j, n-i]\}} | j = 1, \ldots, n-i\}$ to U_{n-i}. Upon receiving $g^{x_1 \ldots x_{i-1} x_{i+1} \ldots x_n}$ each U_i computes the group key $k_i := (g^{x_1 \ldots x_{i-1} x_{i+1} \ldots x_n})^{x_i}$.

Figure 7: Protocol GDH.1 [STW96]

- *Upflow:* In round i, $i = 1, \ldots, n-1$ the user U_i chooses a random $x_i \in \mathbb{Z}_q^*$ and forwards $\{g^{\prod\{x_t | t \in [1,i] \wedge t \neq j\}} | j = 1, \ldots, i\}$ and $g^{x_1 \ldots x_i}$ to U_{i+1}.
- *Broadcast:* In round n the user U_n chooses a random $x_n \in \mathbb{Z}_q^*$ and broadcasts a set $\{g^{\prod\{x_t | t \in [1,n] \wedge t \neq i\}} | i = 1, \ldots, n-1\}$. Upon receiving $g^{x_1 \ldots x_{i-1} x_{i+1} \ldots x_n}$ each U_i computes the group key $k_i := (g^{x_1 \ldots x_{i-1} x_{i+1} \ldots x_n})^{x_i}$.

Figure 8: Protocol GDH.2 [STW96]

- *Upflow:* In round i, $i = 1, \ldots, n-2$ the user U_i chooses a random $x_i \in \mathbb{Z}_q^*$ and forwards $g^{\prod(x_t|t\in[1,i])}$ to U_{i+1}.
- *Broadcast:* In round $n-1$ the user U_{n-1} chooses a random $x_{n-1} \in \mathbb{Z}_q^*$ and broadcasts $g^{\prod(x_t|t\in[1,n-1])}$ to all other users.
- *Response:* In round n the user U_i factors out x_i and sends $g^{\prod(x_t|t\in[1,n-1]\wedge t\neq i)}$ to U_n.
- *Broadcast:* In round $n+1$ the user U_n chooses a random $x_n \in \mathbb{Z}_q^*$ and broadcasts a set $\{g^{\prod(x_t|t\in[1,n]\wedge t\neq i)}|i = 1, \ldots, n-1\}$ to all other users. Upon receiving $g^{x_1 \ldots x_{i-1}x_{i+1}\ldots x_n}$ each U_i computes the secret group key $k_i := (g^{x_1\ldots x_{i-1}x_{i+1}\ldots x_n})^{x_i}$.

Figure 9: Protocol GDH.3 [STW96]

The heuristic security analysis shows that the generic construction is semantically secure against passive adversaries under the DDH assumption. The insecurity against active adversaries comes from the absence of authentication.

For the protocols GDH.2 and GDH.3, Steiner *et al.* proposed two additional extensions that handle join and leave events. In order to proceed with these events in GDH.2 user U_n has to save the contents of the received message during the upflow stage whereas in GDH.3 U_n saves the contents of the first broadcast and response messages. The authors argue that their dynamic extensions preserve semantic security against passive adversaries.

In their subsequent work, Steiner *et al.* [STW98] presented a dynamic group key agreement protocol suite called CLIQUES. It consists of several protocols, that allow the initial key agreement between the founding group members, and auxiliary handling of possible dynamic events (join, leave, group fusion, and subgroup exclusion). In order to proceed with auxiliary protocols for dynamic events each user has to maintain an internal state information. The initial key agreement (IKA) protocol is given by the GDH.2 protocol from [STW96]. For the addition of a group member [STW98] suggests two different protocols that differ in the choice of a controller, i.e., the member who sends the broadcast message enabling other members to update the group key. Further, CLIQUES offers two efficient protocols for the simultaneous addition of multiple members (mass addition), and suggests several forms to process the group fusion event. It can be handled as a special case of the mass join or by the construction of a new super-group via the IKA protocol. Another proposed approach to merge two different groups G_1 and G_2 each having corresponding group keys k_1 and k_2 is to exchange the values g^{k_1} and g^{k_2} and compute the new group key as $k := g^{k_1 k_2}$. Obviously, this approach does not guarantee semantic security with respect to known key attacks. In case of mass exclusion the set

broadcasted by the controller in the last protocol stage does not contain values that would allow excluded members to compute the updated key. In case where a group has to be partitioned into several independent smaller groups each smaller group performs the mass exclusion protocol for all other members.

Security of the CLIQUES protocols has been analyzed based on a snapshot of a current group formation. The protocols do not implicitly provide authentication, and the authors assume that authentic communication channels are used. Therefore, the notion of perfect forward secrecy is not treated, and the adversary is considered to be passive and is represented by a set of all future and former group members with respect to a given snapshot. Thus, the adversary is in possession of all private exponents of these members. Steiner *et al.* address only the issue of key independence and show that the probability of the adversary to distinguish a current group key from a random number is negligible under the `DDH` assumption.

Later, Steiner *et al.* [STW00] extended the CLIQUES suite by another initial key agreement IKA.2 that corresponds to the GDH.3 protocol from [STW96] and a protocol that allows to refresh the group key where one group member generates a fresh private exponent and repeats the last broadcast round of the original IKA.1 or IKA.2 protocol using the updated values. The authors subsequently repeat their heuristic proof from [STW98] to show that the extended CLIQUES suite is semantically secure against passive adversaries under the `DDH` assumption. The authors also claim that the protocol is contributory. This holds only if the adversary is not allowed to reveal private exponents of honest participants, that is only in the weak corruption model.

2.3.1 A Variant by Manulis

Manulis [Man05] describes an elliptic curve variant of the initial key agreement protocol of CLIQUES and its dynamic extensions achieving a better trade-off between computation and communication costs, and analyzes the deployment of the protocol suite in mobile ad-hoc group communication scenarios. The key secrecy of this modification has been argued intuitively based on the `ECDDH` assumption [BSS05].

2.4 Protocols by Ateniese, Steiner, and Tsudik

Ateniese *et al.* [AST98] proposed two authenticated group key agreement protocols, A-GDH.2 and SA-GDH.2, based on the modifications of the GDH.2 protocol from [STW96]. In the proposed protocols every user U_i holds a corresponding long-term key pair (sk_i, g^{sk_i}). The protocol A-GDH.2 proceeds during its first stage similar to GDH.2 but in the last stage U_n broadcasts a set $\{g^{K_{in}\prod\{x_t|t\in[1,n]\wedge t\neq i\}}|i=1,\ldots,n-1\}$ where $K_{in} := F(g^{sk_i sk_n})$

with $F()$ either a reduction modulo q or a cryptographic hash function with domain $\{0,1\}^*$ and image $\mathbb{Z}_q^*$ where q is the order of $\mathbb{G}$. At the end of the protocol every user U_i computes the secret group key $k := (g^{K_{in}\prod\{x_t|t\in[1,n]\wedge t\neq i\}})^{K_{in}^{-1}x_i}$. In this form the authentication is performed indirectly via the controller U_n. Ateniese *et al.* provide a heuristic security analysis of A-GDH.2. They argue that the protocol is resistant against known-key attacks, and provides implicit authentication and perfect forward secrecy in the presence of passive adversaries. As for the active adversaries the authors point out that some attacks against the semantic security of the protocol are possible due to the missing key confirmation property, i.e., it is possible for the active adversary to share a group key with a subset of group members. Also, the implicit authentication in A-GDH.2 is given in a weak form, since there is no direct authentication between the members, but the controller which is assumed to be trusted authenticates himself to all other members. The authors point out that the protocol is susceptible to the attacks by dishonest participants wishing to alter the group formation during the protocol execution by excluding (or skipping) some of its participants. Thus, the protocol does not provide key confirmation in case that some of its participants are dishonest.

In order to prevent some of the described attacks, Ateniese *et al.* proposed a modified protocol version, called SA-GDH.2, with the intention to achieve the informally defined notion of a complete group key authentication, i.e., any two members compute the same group key only if every member has contributed to its computation. The protocol proceeds as described in Figure 10.

Ateniese *et al.* informally argue that SA-GDH.2 provides complete group key authentication and is resistant against known-key attacks in the presence of active adversaries. Further, the authors claim that both protocols can be easily extended to provide key confirmation by including a value $g^{F(k_n)}$ where k_n is the group key computed by U_n into the last message broadcasted by U_n such that each U_i who receives this message is able to verify whether $g^{k_i} \stackrel{?}{=} g^{k_n}$ holds.

In their subsequent work Ateniese *et al.* [AST00] used the ideas from [AST98] to add authentication to the protocols for the initial key agreement and handling of dynamic events of the CLIQUES suite from [STW98]. However, later Pereira and Quisquater [PQ01, PQ03a] discovered some attacks against implicit key authentication, perfect forward secrecy, and resistance against known-key attacks of A-GDH.2 and its dynamic extensions, as well as attacks against complete group key authentication of SA-GDH.2 protocol in the presence of an active adversary. Note that these attacks do not concern the security of the original CLIQUES protocols in [STW98] that remain semantically secure against passive adversaries (in case of weak corruptions).

- *Upflow:* In round i, $i = 1, \ldots, n-1$ the user U_i receives a set of n intermediate values $\{V_t | t = 1, \ldots, n\}$ with

$$V_t = \begin{cases} g^{\frac{x_1 \cdots x_{i-1}}{x_t} \cdot K_{t1} \cdots K_{t(i-1)}} & \text{if } t \leq i-1 \\ g^{x_1 \cdots x_{i-1} \cdot K_{t1} \cdots K_{t(i-1)}} & \text{if } t > i-1, \end{cases}$$

  updates each value as follows

$$V'_t = \begin{cases} V_t^{K_{it} x_i} = g^{\frac{x_1 \cdots x_i}{x_t} \cdot K_{t1} \cdots K_{ti)}} & \text{if } t < i \\ V_t^{K_{it} x_i} = g^{x_1 \cdots x_i \cdot K_{t1} \cdots K_{t(i)}} & \text{if } t > i \\ V_t & \text{if } t = i, \end{cases}$$

  and forwards the updated set $\{V'_t | t = 1, \ldots, n\}$ to U_{i+1}. Note that U_1 starts his computation with an empty set and defines $V'_1 := g$.

- *Broadcast:* In round n the user U_n chooses a random $x_n \in \mathbb{Z}_q^*$, updates received $\{V_t | t = 1, \ldots, n\}$ as described above, and broadcasts $\{V'_t | t = 1, \ldots, n\}$ to all other users. Upon receiving the message each U_i selects the appropriate V'_i and computes the group key as $k := (V'_i)^{x_i \cdot K_{1i}^{-1} \cdots K_{ni}^{-1}} = g^{x_1 \cdots x_n}$.

Figure 10: Protocol SA-GDH.2 [AST98]

2.5 Protocol by Steer, Strawczynski, Diffie, and Wiener

The protocol proposed by Steer *et al.* [SSDW90] to secure audio teleconference systems is the earliest protocol that computes the group key using the structure of a linear tree. Although, the authors do not mention the tree structure explicitly, the mathematical structure of the computed group key is similar to the one obtained from a linear tree. All operations are performed in a cyclic group $\mathbb{G}$ of prime order p generated by g. It is assumed that all users have public-key certificates generated by a trusted party that can be used to sign messages. The protocol proceeds as described in Figure 11. Note that X_n has the algebraic form $g^{x_n g^{x_{n-1} g^{\cdots g^{x_3 g^{x_1 x_2}}}}}$. The authors also describe an efficient addition mechanism for new members by considering a new member as U_{n+1} and appending his input y_{n+1} to the accumulated chain calculation as in the second stage of the protocol. However, this mechanism is not semantically secure against known-key attacks. The authors claim that the protocol provides key secrecy and security against impersonation attacks. However, they do not give any security analysis with respect to either a passive or an active adversary.

- In round 1 each U_i chooses a random $x_i \in_R \mathbb{Z}_p$, and broadcasts $y_i := g^{x_i}$. Upon receiving these values all users get indices according to the ordered list of their identities, i.e., $U_1, \ldots, U_n$, and U_1 computes $X_{i+1} := y_{i+1}^{X_i}$ for all $i = 1, \ldots, n-1$ starting with $X_1 = x_1$.
- In round i, $i = 2, \ldots, n-1$ user U_i, $i = 2, \ldots, n-1$ receives Y_{i-1} (U_2 uses $Y_1 := y_1$), computes $X_i := Y_{i-1}^{x_i}$, $Y_i := g^{X_i}$, broadcasts Y_i to all other users, and computes $X_{j+1} := y_{j+1}^{X_j}$ for all $j = i, \ldots, n-1$.
- In round n all users learn X_n and use it to derive the group key k. Each U_i broadcasts own certificate. Upon receiving all certificates U_i verifies each of them.
- In round $n+1$ each U_i signs a hash value of $(y_1, \ldots, y_n)$ and broadcasts it to other users. Upon receiving all messages each U_i verifies the signature and the hash value.

Figure 11: Protocol by Steer, Strawczynski, Diffie, and Wiener [SSDW90]

2.6 Protocol by Becker and Wille

Becker and Wille [BW98] proposed two static group key exchange protocols, called Octopus and Hypercube. Although, the protocols do not assign users to the leaf nodes of a tree, the algebraic structure of the computed group key is similar to the one that can be obtained from a balanced binary tree. The main building block of the Octopus protocol is a four-party key agreement described in Figure 12. In the first round U_0 and U_1 in parallel

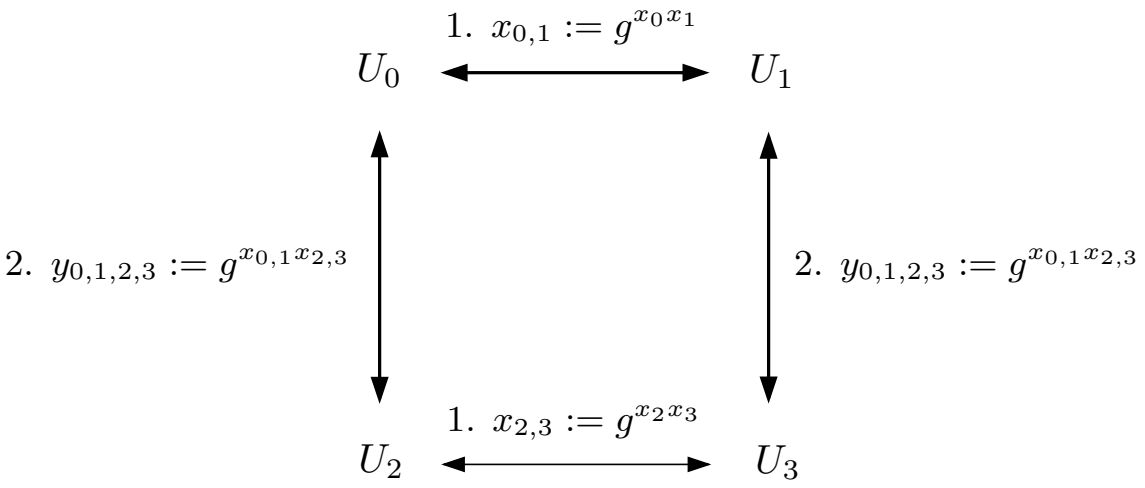

Figure 12: Example: Main Building Block of Protocols Octopus and Hypercube[BW98]

with U_2 and U_3 compute $x_{0,1} := g^{x_0 x_1}$ respectively $x_{2,3} := g^{x_2 x_3}$, respectively. In the second round U_0 and U_2 in parallel with U_1 and U_3 compute the resulting shared key $y_{0,1,2,3} := g^{x_{0,1} x_{2,3}} = g^{g^{x_0 x_1} g^{x_2 x_3}}$. In the Octopus protocol all users are ordered into four subgroups of almost equal sizes in which one user, denoted U_i with $i \in \{0, \ldots, 3\}$, takes the

role of a controller (similar to the role of the sponsor in [KPT01]). The protocol consists of the three stages described in Figure 13. For the case where $n = 2^d$, $d \in \mathbb{N}$ Becker and

- *Stage 1:* Each subgroup controller U_i, $i \in \{0, \ldots, 3\}$, computes an individual Diffie-Hellman key with each of the subgroup members. By $x_{i,j}$ we denote the secret key shared between U_i and the j-th member of the i-th subgroup.
- *Stage 2:* Each controller U_i computes $\hat{x}_i := \prod_j x_{i,j}$ and uses it in the protocol from Figure 12 to compute the group key $k := g^{g^{\hat{x}_1\hat{x}_2} g^{\hat{x}_3\hat{x}_4}}$.
- *Stage 3:* Each controller U_i computes $(g^{\hat{x}_{(i+2) \bmod 4}})^{\hat{x}_i / x_{i,j}}$ (using the received value $g^{\hat{x}_{(i+2) \bmod 4}}$ from Stage 2) and sends this together with another received value $g^{g^{\hat{x}_{(i+1) \bmod 4} \hat{x}_{(i+3) \bmod 4}}}$ to the j-th member of the i-th subgroup who uses $x_{i,j}$ to compute $g^{\hat{x}_{(i+2) \bmod 4} \hat{x}_i}$ and k.

Figure 13: Protocols Octopus and Hypercube [BW98]

Wille proposed a Hypercube protocol where all users are arranged into a d-dimensional hypercube, i.e., a graph in the form of a cube with 2^d vertices, each of them connected to d other vertices. It is assumed that for each user U_i there is a label i of d bits with $i \in \mathbb{Z}_n$. The protocol proceeds in d communication rounds such that in the j-th round each user U_i performs a Diffie-Hellman key exchange with the user $U_{i \oplus 2^{j-1}}$ using the key computed in the previous round $j - 1$, i.e., in the j-th round users along the j-th dimension of the hypercube compute the shared Diffie-Hellman key. After a total number of d rounds and dn unicast messages all users agree on a group key k. The protocol remains efficient if n equals to a power of two. For the opposite case [BW98] suggests a mixed solution, called 2^d-Octopus, which is based on a combination of the Hypercube and Octopus protocols. Becker and Wille prove the semantic security of their protocols against passive adversaries under the DDH assumption using a heuristic method similar to [STW96]. Both protocols as described in [BW98] do not provide any form of authentication and are, therefore, insecure against impersonation attacks.

2.6.1 A Variant by Asokan and Ginzboorg

Asokan and Ginzboorg [AG00] adopted the password-based authentication to the protocol of Becker and Wille for the scenarios of small ad-hoc groups. They also described a solution for a user to find a partner for the Diffie-Hellman key exchange if the round partner of this user is faulty (this can be seen as a step towards denial of service attacks). Although, the authors described the desirable security properties for their protocol, i.e., group key secrecy, perfect forward secrecy, contributiveness, as well as the tolerance against disruption attempts in the mobile ad-hoc setting, they do not give any security analysis

of these issues.

2.7 Protocols by Kim, Perrig, and Tsudik

Perrig [Per99] designed a static group key exchange protocol that outputs group keys with the same algebraic structure as in the Hypercube protocol using the logical structure of a balanced binary tree T from Figure 3. In the following description by $T_{\langle l,v\rangle}$ we denote a subtree of T rooted at node $\langle l,v\rangle$, and for any non-leaf node $x_{\langle l,v\rangle} := g^{x_{\langle l+1,2v\rangle}x_{\langle l+1,2v+1\rangle}}$ holds. The protocol proceeds as described in Figure 14.

- In round 1 each $U_{\langle l,v\rangle}$, $0 \le v \le 2^l - 1$ randomly chooses $x_{\langle l,v\rangle} \in_R \mathbb{Z}_p$ and broadcasts $y_{\langle l,v\rangle} := g^{x_{\langle l,v\rangle}}$ to every other user.
- In round i, $i = 2, \ldots, d_T+1$ each $U_{\langle l,v\rangle}$, $l > d_T+1-i$, computes $x_{\langle d_T+1-i,\lfloor v/2^{i-1}\rfloor\rangle}$. If $i \neq d_T+1$ then for each subtree $T_{\langle d_T+1-i,v\rangle}$, $0 \le v \le 2^{d_T+1-i}-1$, a user assigned to one of the leaf nodes of $T_{\langle d_T+1-i,v\rangle}$ broadcasts $y_{\langle d_T+1-i,\lfloor v/2^{i-1}\rfloor\rangle} := g^{x_{\langle d_T+1-i,\lfloor v/2^{i-1}\rfloor\rangle}}$ to every other user.

Figure 14: Protocol by Perrig [Per99]

Obviously, at the end of the protocol each user computes $x_{\langle 0,0\rangle}$. This value is then used to derive the group key as $k := H(x_{\langle 0,0\rangle})$.

Kim *et al.* [KPT00, KPT04b] extended Perrig's protocol to the TGDH (for Tree-based Group Diffie-Hellman) protocol suite that handles various dynamic group changes. For this purpose each user $U_{\langle l,v\rangle}$ has to store the structure of T including the public keys of all nodes and the own secret key path $\mathcal{X}_{\langle l,v\rangle}$. TGDH changes the initial tree structure and updates the group key with respect to the changes in the group formation. The authors also propose a policy which keeps the updated tree mostly balanced. Informal security analysis of the protocol in [KPT04b] provides arguments for the issues of key independence and group key secrecy. It focuses on the semantic security against passive adversaries under the DDH assumption. TGDH does not provide implicit authentication or key confirmation. Also, perfect forward secrecy is not considered because of the absence of any long-term keys.

Kim, Perrig, and Tsudik [KPT01, KPT04a] proposed a dynamic extension of the Steer *et al.*'s protocol, which they have called the STR protocol (Figure 15). They applied a linear tree structure for the computation of the group key, and have extremely increased the communication efficiency. We use the label-based notation from Figure 3 to describe their protocol. It is assumed that each user $U_{\langle l,v\rangle}$ is assigned to a leaf node of a linear binary tree T, i.e., v is either 0 or 1. All computations are performed in a special multiplicative cyclic group $\mathbb{G} = \langle g\rangle$ of prime order q where the group operation can be used to derive a

bijection from $\mathbb{Z}_q$ to $\mathbb{Z}_q$.

- In round 1 each $U_{\langle l,v\rangle}$, $1 \leq l \leq n-1$, $v \in \{0,1\}$, randomly chooses $x_{\langle l,v\rangle} \in_R \mathbb{Z}_q^*$, and broadcasts $y_{\langle l,v\rangle} := g^{x_{\langle l,v\rangle}}$ to every other user.
- In round 2 users $U_{\langle n-1,0\rangle}$ and $U_{\langle n-1,1\rangle}$ compute $\mathcal{X}_{\langle n-2,0\rangle} := \{x_{\langle l-1,0\rangle} := y_{\langle l,1\rangle}^{x_{\langle l,0\rangle}} | \forall\ n-1 \geq l \geq 1\}$ ($U_{\langle n-1,1\rangle}$ starts the computation with $y_{\langle n-1,0\rangle}$). Then, $U_{\langle n-1,0\rangle}$ computes and broadcasts $\mathcal{Y}_{\langle n-2,0\rangle} := \{y_{\langle l,0\rangle} := g^{x_{\langle l,0\rangle}} |\ \forall\ x_{\langle l,0\rangle} \in \mathcal{X}\}$. Then, each $U_{\langle l,1\rangle}$, $1 \leq l \leq n-2$ computes $\mathcal{X}_{\langle l-1,0\rangle} := \{x_{\langle l-1,0\rangle} := y_{\langle l,0\rangle}^{x_{\langle l,1\rangle}}\} \cup \{x_{\langle j-1,0\rangle} := y_{\langle j,1\rangle}^{x_{\langle j,0\rangle}} | \forall\ l-1 \geq j \geq 1\}$.

Figure 15: Protocol STR [KPT04a]

Note that each $U_{\langle l,v\rangle}$ learns $x_{\langle 0,0\rangle} = g^{x_{\langle 1,1\rangle} g^{x_{\langle 2,1\rangle} g^{\cdots g^{x_{\langle n-2,1\rangle}} g^{x_{\langle n-1,1\rangle} x_{\langle n-1,0\rangle}}}}}$ after the execution of the protocol. Kim *et al.* suggested to derive the group key using a cryptographic hash function, i.e., $k := H(x_{\langle 0,0\rangle})$. The authors also propose efficient operations to deal with dynamic group changes. In order to handle these events the protocol requires from each user to save the whole structure of T including all public keys $y_{\langle l,v\rangle}$, $1 \leq l \leq n-1$, $v \in \{0,1\}$, and the secret key path of the user's leaf node. STR updates the tree and the group key with respect to the changes of the group formation. In [KPT04a], Kim *et al.* intuitively argue that the protocol provides key independence with respect to the known-key attacks under the DDH assumption. The protocol does not provide implicit authentication, however, the authors assume that all communication channels are authentic. Also, no forward secrecy is considered due to the absence of long-term keys.

2.7.1 Variants by Liao, Manulis, and Schwenk

For completeness we mention that Schwenk *et al.* [SMS01] proposed a protocol suite for multimedia communications which is quite similar to that of TGDH but developed independently and patented [SD].

Manulis [Man05] briefly described elliptic curve variants of the TGDH and STR protocol suites in the context of mobile ad-hoc communication which allows to achieve a better trade-off between computation and communication costs of the protocols. The heuristic analysis of the proposed protocols shows that their semantic security against passive adversaries relies on the ECDDH assumption [BSS05].

Recently, Liao and Manulis [LM06] proposed a tree-based framework for group key agreement in ad-hoc networks, called TFAN. The framework consists of a protocol which can be seen as a combination of the optimized elliptic curve variants of TGDH and STR protocols from [KPT01, KPT04a]. The heuristic security proof shows that semantic security of the protocol relies on the ECDDH assumption.

2.8 Protocol by Lee, Kim, Kim, and Ryu

In [LKKR03], Lee *et al.* extend the TGDH protocol suite [KPT00, KPT04b] using Joux' protocol. All members are assigned to the leaf nodes of a ternary tree T. Obviously, every node of T may be a leaf node, a parent node of two, or a parent node of three child nodes. In case that a non-leaf node $\langle l, v\rangle$ is a parent of three child nodes $\langle l+1, 3v+i\rangle$, $i = 0, 1, 2$, its secret key is computed as $x_{\langle l,v\rangle} := H_1(\hat{e}(P,P)^{x_{\langle l+1,3v\rangle} x_{\langle l+1,3v+1\rangle} x_{\langle l+1,3v+2\rangle}})$ using the computations of the Joux' protocol with a cryptographic hash function $H_1 : \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$. In case that $\langle l, v\rangle$ is a parent of two child nodes $\langle l+1, 3v+i\rangle$, $i = 0, 1$ its secret key is computed as $x_{\langle l,v\rangle} := H_2(x_{\langle l+1,3v\rangle} y_{\langle l+1,3v+1\rangle}) = H_2(x_{\langle l+1,3v+1\rangle} y_{\langle l+1,3v\rangle}) = H_2(x_{\langle l+1,3v\rangle} x_{\langle l+1,3v+1\rangle} P)$ using the computations of an elliptic curve equivalent of the two-party Diffie-Hellman protocol with a cryptographic hash function $H_2 : \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$. The protocol proceeds as described in Figure 16. At the end of the protocol each user computes

- In round 1 each $U_{\langle l,v\rangle}$, $0 \leq v \leq 3^l - 1$ randomly chooses $x_{\langle l,v\rangle} \in_R \mathbb{Z}_q^*$ and broadcasts $y_{\langle l,v\rangle} := x_{\langle l,v\rangle} P$ to every other user.
- In round i, $i = 2, \ldots, d_T + 1$ each $U_{\langle l,v\rangle}$, $l > d_T + 1 - i$, computes $x_{\langle d_T+1-i, \lfloor v/3^{i-1}\rfloor\rangle}$. If $i \neq d_T + 1$ then for each subtree $T_{\langle d_T+1-i,v\rangle}$, $0 \leq v \leq 3^{d_T+1-i} - 1$, a user assigned to one of the leaf nodes of $T_{\langle d_T+1-i,v\rangle}$ broadcasts $y_{\langle d_T+1-i, \lfloor v/3^{i-1}\rfloor\rangle} := x_{\langle d_T+1-i, \lfloor v/3^{i-1}\rfloor\rangle} P$ to every other user.

Figure 16: Protocol by Lee, Kim, Kim, and Ryu [LKKR03]

the group key $k := x_{\langle 0,0\rangle}$. Dynamic operations are handled similarly to those of TGDH by updating the group key and the tree structure which is kept mostly balanced. Each user $U_{\langle l,v\rangle}$ is supposed to save own secret key path $\mathcal{X}_{\langle l,v\rangle}$ and the structure of T including all public keys to process dynamic events. The protocol does not provide authentication. Lee *et al.* specify a special assumption which they call a Decisional Ternary Tree Group Bilinear Diffie-Hellman (`DTGBDH`) assumption (which is polynomial-time reducible to the decisional version of the `BDH` assumption) and apply a heuristic security proof (similar to the one from [KPT04b]) to show that under this assumption a passive adversary is not able to distinguish $\hat{e}(P,P)^{x_{\langle 1,0\rangle} x_{\langle 1,1\rangle} x_{\langle 1,2\rangle}}$ from a random number. However, the proof does not consider hash functions H_1 and H_2 used to derive the secret keys of the tree nodes. Beside that, the protocol provides neither implicit key authentication nor key confirmation nor security against key control attacks.

2.9 Protocols by Barua, Dutta, and Sarkar

The protocol proposed by Barua *et al.* [BDS03] uses a similar approach as Lee *et al.* to extend the Joux' protocol to the group setting. Barua *et al.* described a top down recursive

procedure which constructs a balanced ternary tree down to the $(d_T - 1)th$-level. All nodes at level $d_T - 1$ have either one, two, or three child nodes, and all nodes at level $l < d_T - 1$ are either leaf nodes or have three child nodes. In general, the protocol proceeds as in Figure 16 with two main differences. First, [BDS03] replaces the elliptic curve equivalent of the two-party Diffie-Hellman key exchange protocol used in [LKKR03] for the parent nodes with two child nodes by the computations according to the Joux' protocol whereby one of the two users, say $U_{\langle d_T,0\rangle}$, simulates the third user by choosing two secret keys $x_{\langle d_T,0\rangle}$, and $x'_{\langle d_T,0\rangle}$, so that both real users can compute $\hat{e}(P,P)^{x_{\langle d_T,0\rangle}x'_{\langle d_T,0\rangle}x_{\langle d_T,1\rangle}}$. Similar to [LKKR03] the protocol uses hash functions to map the secrets of non-leaf nodes to $\mathbb{Z}_q^*$. The second difference is that, unlike [LKKR03], the sponsor does not broadcast the node's public key, but sends it directly to the users in the subtree(s) rooted at sibling node(s). In addition to the unauthenticated protocol [BDS03] describes an authenticated version which is based on the protocol by Zhang *et al.* [ZLK02] which in turn adapts ID-based authentication to the original Joux' protocol. Barua *et al.* also described two operations to handle joins and leaves of users. The heuristic security analysis of the protocol uses the Decisional Hash Bilinear Diffie-Hellman (`DHBDH`) assumption which is related to the non-standard Hash Decisional Diffie-Hellman (`HDDH`) assumption described in [ABR01] which in turn is strictly stronger than `DDH`. The analysis considers only semantic security against passive adversaries and implicit key authentication. In [PQ03b], Pereira and Quisquater described a successful replay attack against the authenticated version of [BDS03].

3 Provably Secure Group Key Exchange Protocols

3.1 Protocol by Katz and Yung

Katz and Yung [KY03] proposed a static group key exchange protocol (as a modification of an earlier heuristically analyzed protocol by Burmester and Desmedt [BD94, BD05]). All mathematical operations are performed in the cyclic group $\mathbb{G} = \langle g \rangle$ of prime order q such that $\mathbb{G}$ is a subgroup of a cyclic group of prime order $p = \beta q + 1$, $\beta \in \mathbb{N}$ where the `DDH` assumption holds. Figure 17 describes an unauthenticated version of the protocol. Note that after the protocol is completed every user holds the same group key

- In round 1 each U_i chooses a random $r_i \in_R \mathbb{Z}_q$ and broadcasts $z_i := g^{r_i}$.
- In round 2 each U_i broadcasts $X_i := (z_{i+1}/z_{i-1})^{r_i}$. Then each U_i computes $k_i := (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i+n-2}$ for $i = 1, \ldots, n$.

Figure 17: Protocol by Katz and Yung [KY03]

$k = k_i = g^{r_1 r_2 + r_2 r_3 + \ldots + r_n r_1} \pmod p$. Additionally, Katz and Yung proposed an authentication compiler based on digital signatures. This compiler requires one additional communication round for participants to agree on a list of nonces that is then used to signature generation of all outgoing messages and verification of all incoming messages. In the security analysis Katz and Yung show that their protocol is AKE-secure if the DDH assumption holds. Due to the limitations of the used security model (denoted KY) the provided security proof does not consider strong corruptions. Also the protocol does not provide mutual authentication and key confirmation. Bohli *et al.* [BVS05] described an attack against key confirmation of the authenticated version of the protocol in the presence of an adversary $\mathcal{A}$ represented by malicious participants. The attack is successful for all $n > 3$, and proceeds as follows: $\mathcal{A}$ corrupts users U_1 and U_3 and continues with the protocol execution according to its specification up to round 3 (corresponds to round 2 in the unauthenticated version). Then $\mathcal{A}$ swaps X_1 and X_3, i.e., it broadcasts $X_1 := (z_4/z_2)^{r_3}$ and $X_3 := (z_2/z_4)^{r_1}$ (instead of original $X_1 := (z_2/z_4)^{r_1}$ and $X_3 := (z_4/z_2)^{r_3}$). Due to the absence of the key confirmation uncorrupted users U_2 and U_4 compute different group keys $k_2 \neq k_4$. This can be seen by computing the quotient $\frac{k_2}{k_4} = X^n \cdot \left(\frac{z_2}{z_4}\right)^{nr_3} \neq 1$. For example, if $n = 4$ then $k_2 = z_1^{4r_2} \cdot X_2^3 \cdot X_1^2 \cdot X_4 = \frac{z_2^{3r_1} \cdot z_3^{3r_2}}{z_4^{r_1} \cdot z_3^{r_4}}$ and $k_4 = z_3^{4r_3} \cdot X_4^3 \cdot X_3^2 \cdot X_2 = \frac{z_4^{3r_3} \cdot z_1^{3r_4}}{z_2^{r_3} \cdot z_1^{r_2}}$. Note that $z_i^{r_j} = z_j^{r_i}$ for any $i \neq j$ with $1 \leq i, j \leq n$. Therefore, $\mathcal{A}$ who acts on behalf of U_1 and U_3 is able to compute $k_2 := \frac{z_2^{3r_1} \cdot z_2^{3r_3}}{z_4^{r_1} \cdot z_4^{r_3}}$ and $k_4 := \frac{z_4^{3r_3} \cdot z_4^{3r_1}}{z_2^{r_3} \cdot z_2^{r_1}}$. In order to prevent this attack Bohli *et al.* suggested that every user U_i before computing the group key k_i checks whether $\prod_i X_i \stackrel{?}{=} 1$ holds.

Additionally, we present an attack whereby a malicious participant U_i being in the strong corruption model is able to control the resulting value of the group key. The attack proceeds as follows: U_i chooses some $\tilde{r} \in \mathbb{Z}_q$ before the execution is started and his goal is to influence other users to accept with the key $\tilde{k} := g^{\tilde{r}}$. During the execution of the protocol U_i waits for all contributions $z_{j \neq i}$ (the common assumption is that communication channel is asymmetric) and reveals internal states of all other participants that include their private exponents $r_{j \neq i}$. Then, U_i uses

$$r_i := \frac{\tilde{r} - (r_1 r_2 + \ldots + r_{i-2} r_{i-1} + r_{i+1} r_{i+2} + \ldots + r_n r_1)}{r_{i-1} + r_{i+1}}$$

to compute own contribution z_i. It is easy to check that $k := g^{r_1 r_2 + r_2 r_3 + \ldots + r_n r_1} = g^{\tilde{r}} = \tilde{k}$ holds.

3.2 Protocol by Abdalla, Bresson, Chevassut, and Pointcheval

Abdalla *et al.* [ABCP06] described a variant of the KY protocol where authentication is achieved by the means of the password-based encryption using a secure symmetric encryption scheme $\mathcal{E} := (\texttt{Gen}, \texttt{Enc}, \texttt{Dec})$ modeled as an ideal cipher [Sha49, DP06], and three functions H_1, H_2, and $Auth$ modeled as random oracles [BR93]. Each participant U_i holds a secret password pw which is common for the whole group. The protocol proceeds as described in Figure 18. The authors prove the AKE-security of their protocol using the

- In round 1 each U_i chooses a random nonce N_i and broadcasts (U_i, N_i).
- In round 2 each U_i computes a session identifier $S := U_1|N_1|\ldots|U_n|N_n$ and its symmetric key $pw_i := H_1(S, i, pw)$. Each U_i chooses a secret exponent $r_i \in_R \mathbb{Z}_q$ and broadcasts $z_i^* := \mathcal{E}.\texttt{Enc}(pw_i, z_i)$ where $z_i := g^{r_i}$.
- In round 3 each U_i decrypts $z_{i-1} := \mathcal{E}.\texttt{Dec}(pw_{i-1}, z_{i-1}^*)$ and $z_{i+1} := \texttt{Dec}(pw_{i+1}, z_{i+1}^*)$, and broadcasts $X_i := (z_{i+1}/z_{i-1})^{r_i}$. Then each U_i computes the temporary key $k_i := (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i+n-2}$ for $i = 1, \ldots, n$, and broadcasts a confirmation token $Auth_i := Auth(S, \{z_j^*, X_j\}_j, k_i, i)$. Then, each user receives and verifies all confirmation tokens and (if all verifications are successful) accepts with the session group key $K_i := H_2(S, \{z_j^*, X_j, Auth_j\}_j, k_i)$.

Figure 18: Protocol by Abdalla, Bresson, Chevassut, and Pointcheval [ABCP06]

BCP$^+$ model under the DDH assumption with additional non-standard assumptions of ROM [BR93] and Ideal Cipher Model (ICM) [Sha49, DP06]. Abdalla *et al.* also proved that their protocol resists dictionary attacks unless the adversary is able to test several passwords in one session. They prove it by showing that the advantage of the adversary to break the AKE-security of the protocol grows linearly with the number of messages that have been built by the adversary. Note that password-based protocols cannot achieve security against malicious participants because of the adversary can use the shared password pw to authenticate messages on behalf of other participants. The attack of Bohli *et al.* against Katz and Yung's protocol works obviously in this protocol too. Assume that $n = 4$. By swapping X_1 and X_3 the adversary $\mathcal{A}$ achieves that U_2 and U_4 compute different temporary keys $k_2 \neq k_4$. Then $\mathcal{A}$ sends to U_4 resp. U_2 a forged confirmation token $Auth_2 := Auth(S, \{z_j^*, X_j\}_j, k_4, 2)$ resp. $Auth_4 := Auth(S, \{z_j^*, X_j\}_j, k_2, 4)$ where X_1 and X_3 are swapped. Both users U_2 and U_4 verify corresponding confirmation tokens successfully and believe that $k_2 = k_4$. But in fact their temporary session keys are different so that their session group keys K_2 and K_4 are different too.

3.3 Protocol by Kim, Lee, and Lee

Kim *et al.* [KLL04] proposed a dynamic extension of Katz and Yung's protocol. Although, some of the computation steps in [KLL04] have certain similarity with the protocol in [KY03] the mathematical structure of the computed group key is completely different. In Figure 19 we describe the setup operation of the protocol. Again, all group members $U_1, \ldots, U_n$ are arranged into a circle. All computations are performed in a cyclic multiplicative group $\mathbb{G}$ of prime order p generated by g. The protocol uses a cryptographic hash function $H : \{0,1\}^* \rightarrow \{0,1\}^l$, and a secure digital signature scheme (`Gen`, `Sign`, `Verify`) with each U_i having a corresponding signature key pair (sk_i, pk_i). In order to

- In round 1 each U_i randomly chooses $N_i \in_R \{0,1\}^l$ and $r_i \in_R \mathbb{Z}_p^*$ and computes $z_i := g^{r_i}$. Additionally, U_n computes $H(N_n|0)$. Then, each U_i generates $\sigma_i^1 := \texttt{Sign}(sk_i, z_i|ID|0)$ (resp. U_n generates $\sigma_n^1 := \texttt{Sign}(sk_n, H(N_n|0)|z_n|ID|0))$ where $ID := \{U_1, \ldots, U_n\}$ is a set of identities, and broadcasts σ_i^1 together with z_i (resp. $H(N_n|0)|z_n$).
- In round 2 each U_i verifies the received signatures and halts if this process fails. Otherwise, U_i computes $t_i^L := H(z_{i-1}^{r_i}|ID|0)$, $t_i^R := H(z_{i+1}^{r_i}|ID|0)$, and $T_i := t_i^L \oplus t_i^R$. Additionally, U_n generates $\hat{T} := N_n \oplus t_n^R$. Each U_i generates $\sigma_i^2 := \texttt{Sign}(sk_i, N_i|T_i|ID|0)$ (resp. U_n generates $\sigma_n^2 := \texttt{Sign}(sk_n, \hat{T}|T_n|ID|0))$ and broadcasts the signature σ_i^2 together with $N_i|T_i$ (resp. $\hat{T}|T_n$). Then, each U_i verifies the received signatures and halts if this process fails. Otherwise, each U_i computes $\tilde{t}_{i+1}^R := T_{i+1} \oplus t_i^R$, $\tilde{t}_{i+2}^R := T_{i+2} \oplus \tilde{t}_{i+1}^R$, ..., $\tilde{t}_{i+n-1}^R := T_{i+n-1} \oplus \tilde{t}_{i+n-2}^R$, and checks that $t_i^L \stackrel{?}{=} \tilde{t}_{i+n-1}^R$. Then, each U_i decrypts $\tilde{N}_n := \hat{T} \oplus \tilde{t}_n^R$, checks whether $H(\tilde{N}_n|0) \stackrel{?}{=} H(N_n|0)$, and computes the session group key $k_i := H(N_1|\ldots|N_n|0)$.

Figure 19: Protocol by Kim, Lee, and Lee [KLL04] (Setup Operation)

handle dynamic events each U_i has to store k, $h_i^L := H(z_{i-1}^{r_i}|k|0)$, $h_i^R := H(z_{i+1}^{r_i}|k|0)$, and $X := H(N_n|k|0)$ and erase all other ephemeral data[3]. These values are used to update the circle structure and the group key upon occurring dynamic group changes in a way which is more efficient compared to the new execution of the setup operation. For the detailed description of the dynamic operations we refer to [KLL04].

The authors prove the AKE-security of their protocol against active adversaries under the `CDH` assumption. However, their proof requires, additionally, the non-standard assumptions of ROM. Though the provided proof does not consider strong corruptions the authors claim that their protocol guarantees strong-forward secrecy with respect to the BCP$^+$ model [BCP02a]. Their argumentation is that the stored hash values h_i^L, h_i^R, and X upon being revealed can be used to compute group keys of the subsequent sessions

[3]Such *erasure technique* has also been discussed in [CFIJ99, Sho99, BPR00, BCP02a].

but not of the previous sessions. However, the authors do not provide any formal proof of this claim. Also the authors claim that their protocol resists attacks aimed to control the value of the resulting group key without explaining what do they understand under the key control and without providing any formal proofs for their claims (also because no formal definitions of key control were available at that time). Bohli *et al.* [BVS05] described some impersonation attacks on this protocol for the case that session identifiers are generated via concatenation of the exchanged message flows as in the BCP [BCP01] and KY [KY03] models. As a result of their replay attack two honest participants compute identical session group keys without being partnered. For the successful attack $\mathcal{A}$ must replay σ_1^1 together with z_1 addressed to U_3. The reason for this attack is that U_1 is not directly neighbored with U_3 so that the substitution does not affect the computation of the session group key, i.e., $k_1 = k_3$. This attack is of technical nature since it allows $\mathcal{A}$ to ask a *Reveal* query to U_3 and obtain k_3 while asking *Test* query to U_1 ($\mathcal{A}$ is allowed to proceed like that because U_1 and U_3 are not partnered). Since $k_3 = k_1$ the adversary makes a correct guess in response to its *Test* query with non-negligible probability and breaks, therefore, the AKE-security of the protocol. In order to prevent this technical attack Bohli *et al.* suggested that in the beginning of the second round each user U_i computes a session identifier $\mathtt{sid}_i := H(ID|N_1|\ldots|H(N_n))$ and use it for signature generation and verification in the continuation of the protocol execution.

3.4 Protocols by Barua and Dutta

Dutta and Barua [DB05a] described an extension of the Katz and Yung's protocol towards dynamic groups. The modified unauthenticated setup protocol proceeds as described in Figure 20. The authors prove the AKE-security of their protocol against a passive

- In round 1 each U_i randomly chooses $r_i \in_R \mathbb{Z}_p^*$ and sends $z_i := g^{r_i}$ to U_{i-1} and U_{i+1} (note that $U_0 = U_n$ and $U_{n+1} = U_1$).
- In round 2 each U_i computes $t_i^L := z_{i-1}^{r_i}$, $t_i^R := z_{i+1}^{r_i}$, and broadcasts $X_i := t_i^L / t_i^R$ (note that $t_i^R = t_{i+1}^L$ for $1 \leq i \leq n-1$, $t_n^R = t_1^L$, and $t_{i+n-1}^R = t_i^L$). Then, each U_i computes $\tilde{t}_{i+1}^R := X_{i+1} t_i^R$, $\tilde{t}_{i+2}^R := X_{i+2} \tilde{t}_{i+1}^R$, ..., $\tilde{t}_{i+n-1}^R := X_{i+n-1} \tilde{t}_{i+n-2}^R$, and checks that $t_i^L \stackrel{?}{=} \tilde{t}_{i+n-1}^R$. Then, each U_i computes the session group key $k_i := \tilde{t}_1^R \cdots \tilde{t}_n^R$, the seed $x := H(k_i)$, and saves t_i^L, t_i^R.

Figure 20: Protocol by Dutta and Barua [DB05a] (Unauthenticated Setup Operation)

adversary using a minor modification of the KY model under the DDH assumption.

Additionally, Dutta and Barua described an authenticated version of their protocol where digital signatures are used to sign every protocol message assuming that every

member U_i has a signature key pair (sk_i, pk_i). This authentication approach is similar to the one described by Katz and Yung [KY03] for the only difference that it does not uses nonces. The authors prove that the authenticated version is AKE-secure under the DDH assumption using the same model as for the unauthenticated version. They also prove that the protocol provides the weak form of forward secrecy. However, it seems that there are several inaccuracies in their proof. First, the proof does not consider corrupt queries which reveal long-term keys (in this case sk_i). However, the forward secrecy requirement assumes that the adversary is allowed to reveal these keys. Second, in contrast to the Katz and Yung's technique the protocol by Dutta and Barua does not use nonces as part of the signed messages. Note that nonces (or any other fresh randomness) are essential to resist replay attacks (as considered in [KY03]). Therefore, it is not clear whether the proposed protocol remains secure if the adversary replays a message from some previous session. Indeed, no such attacks are covered by the proof. Moreover, the simulation described in the proof fails if the adversary replays a message as part of its *Send* query, because these queries are answered from the predefined transcripts obtained through execute queries, and, therefore, any unpredictable *Send* query would not be answered.

Additionally, Dutta and Barua described dynamic operations allowing new members to join to the group and current members to leave it. In order to handle these operations efficiently (without restarting the initial protocol) participants use the saved values: $x := H(k_i)$, t_i^L, and t_i^R. Note that these values are considered as part of internal states of protocol participants. For the proof of AKE-security of the dynamic protocol version the authors apply a variant of the BCP model. They also claim that the dynamic protocol version achieves forward secrecy. However, their proof has the same weaknesses as the proof of the static authenticated version. Additionally, recall that the BCP model does not consider strong corruptions. Therefore, it is not clear whether the dynamic protocol by Dutta and Barua still provides AKE-security if strong corruptions are considered. Beside these weaknesses the protocols by Dutta and Barua are also susceptible to the attack against key control in a similar way as described for the protocol by Katz and Yung in Section 3.1.

In [DB06], Dutta and Barua described a variant of Kim *et al.*'s protocol [KLL04] where a password pw shared between all users $U_1, \ldots, U_n$ is used together with three secure symmetric encryption schemes $(\mathtt{Gen}_i, \mathtt{Enc}_i, \mathtt{Dec}_i)$, $i = 1, 2, 3$, for the purpose of authentication instead of the originally used digital signatures. The protocol proceeds as described in Figure 21. To prove the AKE-security of their protocol, Dutta and Barua applied a variant of the model proposed in [BCP02b] together with the non-standard assumptions of ROM [BR93]. The authors also claimed that the proposed protocol is secure

- In round 1 each U_i randomly chooses $N_i \in_R \{0,1\}^l$ and $r_i \in_R \mathbb{Z}_p^*$ and sends $z_i^* := \mathtt{Enc}_1(pw, z_i)$ where $z_i := g^{r_i}$ to U_{i-1} and U_{i+1}.
- In round 2 each U_i extracts z_{i-1} and z_{i+1}, and computes $t_i^L := H(z_{i-1}^{r_i}|ID|0)$ and $t_i^R := H(z_{i+1}^{r_i}|ID|0)$. For $i = 1, \ldots, n-1$ each U_i computes $T_i := t_i^L \oplus t_i^R$ while U_n computes $T_n := N_n \oplus t_n^R$. For $i = 1, \ldots, n-1$ each U_i broadcasts $\mathtt{Enc}_2(pw, N_i|T_i)$ while U_n broadcasts $\mathtt{Enc}_3(pw, T_n)$. Then, each U_i recovers N_n (as in Kim *et al.*'s protocol) and computes the session group key $k_i := H(N_1|\ldots|N_n)$.

Figure 21: Protocol by Dutta and Barua with Password-Based Authentication [DB06]

against off-line dictionary attacks. However, [ABCP06] described an efficient substitution attack on this protocol which allows to mount a successful dictionary attack revealing the shared password pw.

3.5 Protocols by Bresson and Catalano

Bresson and Catalano [BC04] proposed a family of static constant-round authenticated group key exchange protocols based on the following generalized protocol where each user U_i has a pair of private/public keys (sk_i, pk_i) generated by a key generation algorithm of a public key encryption scheme. These protocols derive the group key from the interpolation of secretly shared polynomials. All computations are performed modulo a sufficiently large prime number p. The generalized protocol proceeds as specified in Figure 22. The authors

- In round 1 each U_i chooses random values $s_i, b_{i,1}, \ldots, b_{i,n-1} \in_R \mathbb{Z}_p$, defines $f_i(z) := s_i + b_{i,1}z + \ldots + b_{i,n-1}z^{n-1} \mod p$ and sends $f_i(j)$ to user U_j.
- In round 2 each U_i computes $f(i) := \sum_{j=1}^{n} f_j(i) \mod p$, encrypts $f(i)$ with the public keys of all users, denoted $ENC_j(f(i))$ and sends the corresponding value to U_j.
- In round 3 each U_i decrypts all received values from the previous round, and interpolates them in $\mathbb{Z}_p$ retrieving the secret $f(0) := s_i' = s_1 + \ldots + s_n \mod p$. Then, each U_i computes $k_i' := F_{s_i'}(U_i)$ where F is a pseudo-random function and broadcasts it to other users. Upon receiving these messages from all other users each U_i checks whether $F_{s_i'}(U_j) = k_j'$ holds for all the received values, and if so computes the group key $k := F_{s_i'}(ID)$, where $ID = \{U_1, \ldots, U_n\}$ is the set of identities.

Figure 22: Protocol by Bresson and Catalano [BC04] (Generalized Version)

prove the AKE-security of their protocol under standard assumptions, i.e., the existence of one-way functions, following the requirements of the BCP model [BCP01]. The protocol also achieves key confirmation. Furthermore, the authors provide two realizations based on the El-Gamal and RSA encryption schemes where the function F is instantiated

by a cryptographic hash function. For the ElGamal-based realization the authors show resilience against key control whereby considering its weaker version with honest participants who are assumed to have a biased source of randomness so that a curious adversary tries to gain extra information and break the AKE-security of the protocol. The authors stress that the case where participants are malicious and try to bias the resulting group key deliberately is not considered. Note that this weaker form of key control is achieved in an inefficient way by requiring that each participant chooses in the first step an additional random value r_i, which is then encrypted with the El-Gamal public keys of all other participants and broadcasted. Note also that all protocols proposed in [BC04] are static.

3.6 Protocols by Bresson, Chevassut, Pointcheval, and Quisquater

Bresson *et al.* [BCPQ01] proposed a static group key exchange protocol (as an authenticated extension of Steiner *et al.*'s protocols from [STW96]). Each member U_i is in possession of a long-lived key-pair (sk_i, pk_i) for the digital signature scheme $(\texttt{Gen}, \texttt{Sign}, \texttt{Verify})$. All mathematic operations are performed in a finite cyclic group $\mathbb{G}$ of the prime order q, $|q| = \kappa$ generated by g. The protocol proceeds as described in Figure 23 whereby $ID := \{U_1, \ldots, U_n\}$ is a set of identities of protocol participants and $H : \{0,1\}^* \rightarrow \{0,1\}^\kappa$ is a cryptographic hash function.

- *Upflow stage:* In round $i = 1, \ldots, n-1$ the user U_i chooses a random $x_i \in_R \mathbb{Z}_q^*$, computes $X_i := \{g^{\prod\{x_t | t \in [1,i] \wedge t \neq j\}} | j = 1, \ldots, i\}$ and $Z_i := g^{x_1 \cdots x_i}$, generates $\sigma_i := \texttt{Sign}(sk_i, ID|X_i|Z_i)$ and forwards $X_i|Z_i$ and σ_i to U_{i+1}. Upon receiving the corresponding message U_{i+1} verifies the attached signature and halts if this verification is not successful.

- *Downflow stage:* In round n the user U_n chooses a random $x_n \in_R \mathbb{Z}_q^*$ computes $X_n := \{g^{\prod\{x_t | t \in [1,n] \wedge t \neq i\}} | i = 1, \ldots, n-1\}$, $Z_n := Z_{n-1}^{x_n}$, generates $\sigma_n := \texttt{Sign}(sk_n, ID|X_n)$ and broadcasts X_n and σ_n. After the verification of σ_n each U_i, $i \in [1, n-1]$ extracts $Z'_i := g^{x_1 \cdots x_{i-1} x_{i+1} \cdots x_n} \in X_n$ and computes the temporary key $k_i := Z'^{x_i}_i$ while U_n computes $k_n := Z_n$. Finally, each U_i accepts with the session group key $K_i := H(ID|X_n|k_i)$

Figure 23: Protocol by Bresson, Chevassut, Pointcheval, and Quisquater [BCPQ01]

To prove the AKE-security of their protocol Bresson *et al.* specify a Group Computational Diffie-Hellman assumption (GCDH) previously surfaced in [STW96, Bon98], which is polynomial-time reducible to the standard cryptographic assumptions CDH and DDH [BCP02c]. The proof itself is performed in the BCPQ model [BCPQ01] with the additional non-standard assumptions of ROM [BR93]. Due to the limitations of the BCPQ model the proof does not consider strong corruptions. Also the issue of key control is

not considered. We point out that an adversary $\mathcal{A}$ represented by a malicious participant U_j (being in the strong corruption model) can control the value of the temporary key k_i computed by some uncorrupted user U_i. For this purpose $\mathcal{A}$ simply chooses $\tilde{x} \in \mathbb{Z}_q^*$ prior to the execution of the protocol and computes own exponent after having revealed exponents of other participants as $x_j := \frac{\tilde{x}}{\prod_{i \neq j} x_i}$. Also set ID may be known prior to the protocol execution. Obviously, whether $\mathcal{A}$ is able to control the resulting value K_i depends either on the collision-resistance property of H, or on the ability of $\mathcal{A}$ to find appropriate values in X_n such that the input of H corresponds to some value chosen by $\mathcal{A}$ prior to the protocol execution. The protocol in Figure 23 does not provide key confirmation and mutual authentication.

Additionally, Bresson *et al.* describe a mechanism to achieve MA-security (key confirmation and mutual authentication). It consists of one additional communication round where each user U_i after having computed K_i as described above computes and broadcasts $H(K_i, U_i)$. Every other member U_j receives this message and checks whether $H(K_j, U_i) \stackrel{?}{=} H(K_i, U_i)$ holds (note, this implies $K_i \stackrel{?}{=} K_j$). If this verification holds for all participants then each user U_i computes the actual group key as $K_i' := H(K_i, 0)$. The authors prove that in the Random Oracle Model this additional round adds mutual authentication and key confirmation with respect to the definition of MA-security in the BCPQ model while preserving AKE-security. Recall that this definition is flawed. Hence, the provided security proof is no more reliable. It is easy to show that the above approach does not provide key confirmation in the presence of malicious participants. This is because the hash value $H(K_i, U_i)$ does not provide sender identification.

Bresson, Chevassut, and Pointcheval [BCP01] extended the static protocol from [BCPQ01] to handle additions (join protocol) and exclusions (remove protocol) of group members. For this purpose every user has to save the last broadcasted set X_n which is then updated with freshly chosen private exponent(s). The remove protocol consists of a single downflow stage where the highest-indexed remaining user U_n deletes from X_n all values addressed to the excluded group members, raises all remaining values to the power of the freshly generated exponent x_n' and broadcasts the updated set X_n'. The join protocol requires an upflow stage where private exponents of joined members are collected in a way described in the setup protocol starting with the position of the highest-indexed member U_n. The downflow stage is similar to that of the setup protocol. For the detailed description of the dynamic operations we refer to [BCP01]. The authors proof the AKE-security of this dynamic protocol in the BCP model [BCP01] using the non-standard assumptions of ROM under the `GCDH` assumption. Their proof is similar to the proof in [BCPQ01] and does not consider strong corruptions. Indeed, it is possible to show that if an adversary $\mathcal{A}$ obtains

private exponents of participants in one protocol sessions then it is able to compute group keys from the previous sessions using public values in X_n. Considerations concerning key control issues are similar to that of the static protocol.

In their subsequent work in [BCP02a], Bresson, Chevassut, and Pointcheval revised the protocols from [BCP01] and proposed a variant that is secure under standard assumptions, i.e., without considering assumptions of ROM. Instead of digital signatures as in [BCP01] the authentication in the protocols from [BCP02a] is carried out by a message authentication code (MAC) function. Each user U_i is in possession of an El-Gamal-like long-lived key (s_i, g^{s_i}) where g is a generator of some group $\mathbb{G}$ where the DDH assumption holds. The MAC-key K_{ij} used for authentication between users U_i and U_j is derived as $F_1(g^{s_i s_j})$ where F_1 is a universal hash function $H_r()$ ([Gol04, Section 6.4.3]) which takes as input beside $g^{s_i s_j}$ an additional random string $r = r_{ij}$ which U_i and U_j receive during the registration of their identities from the Certification Authority which is part of the Public-Key Infrastructure (PKI). Note that K_{ij} does not expose. The group key is derived as $K := F_2(ID, X_n, k)$ where F_2 is a universal hash function $H_r()$ where the required random string $r = r_k$ is chosen by the user U_n who sends the final broadcast message. The authors prove the AKE-security of their protocol under the Group Decisional Diffie-Hellman GDDH and the Multi Decisional Diffie-Hellman (MDDH) assumptions which are polynomial-time reducible to the DDH assumption. It is worth being noticed that the protocol in [BCP02a] does not explicitly provide MA-security. Indeed, if the mechanism from [BCPQ01, BCP01] is applied then the proof requires non-standard assumptions of ROM. As for the requirement of key control we notice that the adversary $\mathcal{A}$ (being in the strong corruption model) represented by a malicious participant U_j can control the value of k (the computation is similar to that of the protocol in [BCPQ01]) and may learn ID prior to the execution of the protocol. In addition to that if U_n is malicious then it can choose the random string $r = r_k$ used in the universal hash function $H_r()$ non-uniformly implying that resulting hash values are not uniformly distributed. Hence, the probability that $\mathcal{A}$ controls the value of the key is given by the probability that for two different input values of F_2, say α and β, the adversary finds r_α and r_β such that $H_{r_\alpha}(\alpha) = H_{r_\beta}(\beta)$. Obviously, this is some non-standard requirement of collision-resistance of universal hash functions.

In their another work in [BCP02b], Bresson *et al.* proposed a static variant of [BCPQ01] with the password-based authentication. The authors show the AKE-security of their protocol in the BCPQ model [BCPQ01] under the Trigon Group Computational Diffie-Hellman (TGCDH) assumption (which is a special form of the GCDH assumption and thus reducible to CDH and DDH) in addition to the non-standard assumptions of ROM. Their

proof also shows that the protocol is resistant against dictionary attacks. For this purpose in addition to private exponents x_i each U_i has a second private exponent ν_i such that the resulting group key is derived as $K := H(ID, X_n, k)$ where H is a cryptographic hash function and $k = g^{\prod_i (x_i \nu_i)}$. However, the protocol does not deal with dynamic group changes, and its proof does not consider the requirement on forward secrecy. The authors also mention that MA-security can be achieved using the hash function based mechanism from [BCPQ01]. However, this mechanism does not guarantee security in the presence of malicious participants.

Recently, Bresson *et. al.* [BCP06] proposed another static password-based group key exchange protocol, called GOKE, for IEEE802.11's ad-hoc mode. This protocol proceeds as described in Figure 24 whereby $VP(x_i)$ (validity proof) is a non-interactive zero-knowledge proof [Sch89] for the knowledge of x_i in the exponent of the elements of X_i, pw_i is a secret password shared between U_n and U_i, f is a symmetric encryption function, and H_1, H_2, H_3 are cryptographic hash functions.

- *Upflow stage:* In round $i = 1, \ldots, n-1$ the user U_i chooses random $x_i \in_R \mathbb{Z}_q^*$, $r_i \in_R \{0,1\}^\kappa$ computes $X_i := \{g^{\prod\{x_t | t \in [1,i] \wedge t \neq j\}} | j = 1, \ldots, i\}$ and $Z_i := g^{x_1 \cdots x_i}$, generates $VP(x_i)$ and forwards $r_i | X_i | Z_i$ and $VP(x_i)$ to U_{i+1}. Upon receiving the corresponding message U_{i+1} verifies the attached validity proof and halts if this verification is not successful.

- *Downflow stage:* In round n the user U_n chooses a random $x_n \in_R \mathbb{Z}_q^*$, computes for all $i = 1, \ldots, n$, the temporary key $k := g^{\prod_i x_i}$, $k_i := K^{1/x_i}$ using elements from X_{n-1}, $k_i' := k_i^{\alpha_i}$ where $\alpha_i \in_R \mathbb{Z}_q^*$, and broadcasts $k_i^* := k_i' \cdot f(pw_i)$. Upon receiving this value each U_i "unmasks" k_i', computes $k_i'' := k_i'^{x_i}$, an authenticator $\texttt{Auth}_i := H_1(r_1 | \ldots | r_n | i | k_i'')$ and sends $\texttt{Auth}_i$ to U_n.

- In round $n+1$ user U_n checks $\texttt{Auth}_i \stackrel{?}{=} H_1(r_1 | \ldots | r_n | i | k^{\alpha_i})$ for each received $\texttt{Auth}_i$. If all received authenticators are valid then U_n broadcasts $\texttt{Auth}_i' := H_2(r_1 | \ldots | r_n | i | k^{\alpha_i} | k_i)$ and k_i.

- In round $n+2$ each U_i checks $\texttt{Auth}_i' \stackrel{?}{=} H_2(r_1 | \ldots | r_n | i | k_i'' | k_i)$. If this verification holds then U_i accepts with the session group key $K_i := H_3(r_1 | \ldots | r_n | k_i^{x_i})$.

Figure 24: Protocol by Bresson, Chevassut, and Pointcheval with Password-Based Authentication [BCP06]

The authors prove the AKE-security of the protocol using a formal setting from [BCP02b] together with the non-standard assumptions of ROM (for the applied hash functions) under the `TGCDH` assumption. Note that in this protocol U_n acts as a sole authenticator and checks whether all participants computed the same key. There is no direct authentication and key confirmation between any two participants U_i and U_j with

$1 \leq i, j < n$. Thus, if the authenticator U_n is malicious then he can mount a successful attack against the mutual authentication and key conformation properties by sending $\texttt{Auth}'_i := H_2(r_1|\ldots|r_n|i|k''_i|\tilde{k}_i)$ with some fake $\tilde{k}_i$.

3.7 Protocols by Dutta, Barua, and Sarkar

In [DBS04], Dutta, Barua, and Sarkar extended their heuristically analyzed unauthenticated protocol from [BDS03] (Section 2.9) by the authentication mechanism based on the non-interactive multi-signature scheme by Boldyreva [Bol03] and on the pairing-based signature scheme by Boneh, Lynn, and Shacham [BLS01]. Recall that the protocol assigns users to the leaf nodes of a balanced ternary key tree T and applies iterations of Joux' protocol to compute the resulting key at the root of T which is then used to derive the session group key. The tree is constructed in such a way that all nodes at level $d_T - 1$ are parent nodes of either one, two, or three child nodes, and all nodes at levels $l < d_T - 1$ are either leaf nodes or parents of exactly three child nodes. In general the protocol proceeds as described in Figure 25 with the following computation rules for the secret values $x_{\langle d_T-1,v\rangle}$ where $H : \mathbb{G}_2 \to \mathbb{Z}_q^*$ is a cryptographic hash function:

- if $\langle d_T - 1, v\rangle$ is a parent of one leaf node then $x_{\langle d_T-1,v\rangle}$ is exactly the secret value chosen by the user assigned to that leaf node,
- else if $\langle d_T - 1, v\rangle$ is a parent of two leaf nodes $\langle d_T, 3v + i\rangle$, $i = 0, 1$ then $x_{\langle d_T-1,v\rangle} = H(\hat{e}(P, P)^{x_{\langle d_T,3v\rangle} x_{\langle d_T,3v+1\rangle} x'})$ where $x_{\langle d_T,3v\rangle}$ and x' are secret values chosen by the user assigned to $\langle d_T, 3v\rangle$, and $x_{\langle d_T,3v+1\rangle}$ is chosen by the user assigned to $\langle d_T, 3v + 1\rangle$,
- else if $\langle d_T - 1, v\rangle$ is a parent of three leaf nodes $\langle d_T, 3v + i\rangle$, $i = 0, 1, 2$ then $x_{\langle d_T-1,v\rangle} = H(\hat{e}(P, P)^{x_{\langle d_T,3v\rangle} x_{\langle d_T,3v+1\rangle} x_{\langle d_T,3v+2\rangle}})$ where each $x_{\langle d_T,3v+i\rangle}$ is chosen by the user assigned to $\langle d_T, 3v + i\rangle$

Note that each node $\langle l, v\rangle$ with $l < d_T - 1$ is either a leaf node or a parent of exactly three nodes. Therefore, $x_{\langle l,v\rangle}$ is either chosen by the user assigned to $\langle l, v\rangle$ or computed as $H(\hat{e}(P, P)^{x_{\langle l+1,3v\rangle} x_{\langle l+1,3v+1\rangle} x_{\langle l+1,3v+2\rangle}})$ via Joux' technique.

The secret value $x_{\langle 0,0\rangle}$ at the root of the tree is then used as the session group key.

Dutta *et al.* prove the AKE-security of their protocol against active adversaries under the non-standard cryptographic assumption DHBDH in the modified version of the KY security model[KY03]. Unlike the authentication procedure in Katz and Yung's protocol the protocol by Dutta *et al.* does not use nonces as part of signed messages. Note that nonces are useful to resist replay attacks. Therefore, it is not clear whether the proposed protocol remains secure in case where an active adversary replays previous messages. The

- In round 1 each $U_{\langle l,v\rangle}$, $0 \leq v \leq 3^l - 1$ randomly chooses $x_{\langle l,v\rangle} \in_R \mathbb{Z}_q^*$, computes and sends $y_{\langle l,v\rangle} := x_{\langle l,v\rangle}P$ together with the corresponding digital signature σ to every user in the subtree(s) rooted at the sibling node(s) of $\langle l, v\rangle$. Every user verifies received signatures before he proceeds with the protocol.
- In round i, $i = 2, \ldots, d_T+1$ each $U_{\langle l,v\rangle}$, $l > d_T+1-i$, computes $x_{\langle d_T+1-i,\lfloor v/3^{i-1}\rfloor\rangle}$. If $i \neq d_T+1$ then for each subtree $T_{\langle d_T+1-i,v\rangle}$, $0 \leq v \leq 3^{d_T+1-i}-1$, a user (sponsor) assigned to one of the leaf nodes of $T_{\langle d_T+1-i,v\rangle}$ computes and sends $y_{\langle d_T+1-i,\lfloor v/3^{i-1}\rfloor\rangle} := x_{\langle d_T+1-i,\lfloor v/3^{i-1}\rfloor\rangle}P$ together with the corresponding non-interactive multi-signature σ to every user in the subtree(s) rooted at the sibling node(s) of $\langle d_T+1-i, v\rangle$. Every user verifies received signatures before he proceeds with the protocol.

Figure 25: Protocol by Dutta, Barua, and Sarkar [DBS04]

simulation described in the proof fails if the adversary replays a message as part of his send query, because send queries are answered from the predefined transcripts obtained through execute queries, and, therefore, any unpredictable send query (such as a replayed message) cannot be answered. Also, the security of some parts of Dutta *et al.*'s modifications to the original BCPQ model [BCPQ01] are arguable as discussed in [Man06, Section 3.8.4]. Beside this, it is possible to show that the protocol is susceptible to the attack against key control (in the strong corruption model). The idea behind the attack is that the adversary represented by a malicious participant may know the tree structure prior to the execution of the protocol and own position within it. The adversary adaptively computes all secret values in its path up to $x_{\langle 0,0\rangle}$ prior to the execution of the protocol. Then, during the protocol execution it influences honest participants that are assigned to the leaf nodes of the subtrees rooted at nodes that are siblings of the nodes in the adversarial path to compute each $x_{\langle l,v\rangle}$ as chosen by the adversary. In the strong corruption model this attack is simple. For example, consider that two honest participants are assigned to the leaf nodes $\langle d_T, 3v+1\rangle$ and $\langle d_T, 3v+2\rangle$ and the malicious participant is assigned to $\langle d_T, 3v\rangle$, and chooses $x_{\langle d_T-1,v\rangle}$ prior to the protocol execution as the output of $H(\hat{e}(P,P)^{\tilde{x}})$ for some chosen $\tilde{x}$. Then during the protocol execution it reveals $x_{\langle d_T,3v+1\rangle}$ and $x_{\langle d_T,3v+2\rangle}$ as part of the internal information of honest participants and computes $x_{\langle d_T,3v\rangle} := \frac{\tilde{x}}{x_{\langle d_T,3v+1\rangle}x_{\langle d_T,3v+2\rangle}}$. Further, this kind of the attack can be performed for all $x_{\langle l,v\rangle}$ in the adversarial path including the session group key $x_{\langle 0,0\rangle}$.

In [DB05b], Dutta and Barua extended the above protocol by additional operations that handle dynamic group changes, i.e., addition and deletion of group members. Both events are handled using a sponsor and result in the updated logical tree T' and the updated secret value at the root of T' whereby some secret values $x_{\langle l,v\rangle}$ remain unchanged. The authentication is achieved using digital signatures and multi-signatures as in [DBS04].

The authors prove the AKE-security of their dynamic protocol against active adversaries under the DHBDH assumption using a mix of the BCP$^+$ [BCP02a] and KY [KY03] models with own technical modifications. The proof considers only weak corruptions. Obviously, no forward secrecy in case of strong corruptions is provided because the knowledge of any unchanged $x_{\langle l,v\rangle}$ can be used to compute the previous value of $x_{\langle 0,0\rangle}$.

4 Summary and Discussion

In Table 4 we summarize results of our security-focused survey of group key exchange protocols while considering only provably secure protocols from Section 3 since security models used in their proofs provide a solid background for a fair comparison of their security states. We consider only protocols that have not been found to be flawed despite of their security proof. For each considered protocol we specify the applied security model together with some possibly used non-standard models like Random Oracle Model (ROM) [BR93] or Ideal Cipher Model (ICM) [Sha49, DP06]. Additionally, we specify the underlying cryptographic assumption, and point out whether the proof considers strong (S) or weak (W) corruptions. In the last columns we give the protocol type (S for static; D for dynamic).

4.1 Strong vs. Weak Corruptions

Observe, only few security proofs of the described protocols consider a powerful adversary which is given access to strong corruptions. From the analysis of security models in [Man06] we know that only the BCP$^+$ [BCP02a] and KS/UC-KS [KS05] models provide definitions that consider strong corruptions. However, there exists no group key exchange protocol proven secure in the KS/UC-KS models. The only protocols proven secure in the BCP$^+$ model have been proposed by Abdalla *et al.* [ABCP06] and by Bresson *et al.* [BCP02a]. The protocol proposed by Abdalla *et al.* is static. Intuitively, all static protocols provide security in the strong corruption model as long as they provide security in the weak corruption model. This is because in static protocols internal (ephemeral) information used for the computation of the group key is chosen independently at random for each new protocol execution (session). However, this is not the case in dynamic protocols. The protocol proposed by Bresson *et al.* [BCP02a] is dynamic. However, it does not provide security (in particular in case of forward secrecy) in the strong corruption model. Therefore, its proof considers only weak corruptions. For the protocol proposed by Dutta *et al.* [DBS04] and its dynamic version in [DB05b] we pointed out that given security proofs do not consider possible replay attacks. For the dynamic protocol proposed by

Table 1: Analysis of Provably Secure Group Key Exchange Protocols

Protocol		Model(s)	Assumption(s)	Corr.	S/D
Abdalla *et al.*	[ABCP06]	BCP$^+$ + ICM, ROM	DDH	S	S
Barua and Dutta	[DB05a]	KY	DDH	W	S
Barua and Dutta	[DB05a]	BCP	DDH	W	D
Barua and Dutta	[DB05b]	BCP	DHBDH	W	D
Bresson and Catalano	[BC04]	BCP	OW	W	S
Bresson *et al.*	[BCPQ01]	BCPQ + ROM	GCDH	W	S
Bresson *et al.*	[BCP01]	BCP + ROM	GCDH	W	D
Bresson *et al.*	[BCP02a]	BCP$^+$	GDDH, MDDH	W	D
Bresson *et al.*	[BCP02b, BCP06]	BCPQ + ROM	TGCDH	W	S
Dutta *et al.*	[DBS04]	BCPQ	DHBDH	W	S
Katz and Yung	[KY03]	KY	DDH	W	S
Kim, Lee, and Lee	[KLL04]	BCP,KY + ROM	CDH	W	D

Kim, Lee, and Lee in [KLL04] security in the strong corruption model has been claimed but not formally proven (also due to the absence of adequate security models at that time).

4.2 Standard vs. Non-Standard Assumptions

Security proofs of the protocols in [ABCP06, BCPQ01, BCP01, KLL04] require non-standard assumptions of ROM and/or ICM. Security of the protocols in [DB05a] and [KY03] is based on the standard cryptographic assumption DDH. Security of the generalized protocol by Bresson and Catalano [BC04] is based on the standard cryptographic assumption concerning the existence of one-way functions (OW). Security of the protocols proposed by Bresson *et al.* in [BCPQ01, BCP01, BCP02b, BCP06] has been proven under the assumptions GCDH and TGCDH which are polynomial-time reducible to the standard cryptographic assumptions CDH and DDH [BCP02c]. The CDH assumption has also relevance

for the security of the protocol proposed by Kim, Lee, and Lee [KLL04]. The protocol proposed by Bresson, Chevassut, and Pointcheval in [BCP02a] relies on the GDDH and MDDH assumptions which are polynomial-time reducible to DDH. The only non-standard cryptographic assumption is DHBDH which is used in the protocols from [DBS04, DB05b].

4.3 Attacks of Malicious Participants

We focus on the attacks of malicious participants against the properties of key confirmation and mutual authentication (MA-security), and key control and contributiveness. The analysis of security models in [Man06] showed that only the KS/UC-KS and BVS models provide definitions considering requirements on key confirmation and mutual authentication with respect to the malicious participants. Intuitively, all password-based authentication protocols including [ABCP06, BCP02b, BCP06] are susceptible to such attacks of malicious participants (see Section 3.2 for an example of the attack against [ABCP06]) because the password-based authentication does not provide identification when used in the group setting. However, such identification is important if all protocol participants must authenticate mutually. Obviously, protocols where mutual authentication is performed via digital signatures are more suitable for this purpose. Still, none of the security proofs of the protocols in [BCPQ01, BCP01, BC04, KY03, KLL04, DBS04, DB05a, DB05b] that apply digital signatures considers this kind of attacks of malicious participants. It is worth being mentioned that fortunately the generic compiler for the security against insider attacks described in [KS05] can be used to provide resistance against mutual authentication and key confirmation attacks of malicious participants in all of the above protocols.

As described in [Man06], none of the security models used in the security proofs of the GKE protocols in Table 4 provides formal definitions concerning key control and contributiveness in case of strong corruptions. This is the reason why none of the currently existing group key exchange protocols could be proven secure against this kind of attacks (see Section 3.1 for an example attack against [KY03] in case of strong corruptions).

4.4 Main Results

One of the most important observations w.r.t. the given analytical survey is that none of the currently available dynamic group key exchange protocols provides security against strong corruptions and malicious participants under standard cryptographic assumptions. We stress that this kind of security is especially challenging for dynamic protocols where participants need to save some secret auxiliary information in order to update the session group key on occurring dynamic group changes.

References

[ABCP06] Michel Abdalla, Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Password-Based Group Key Exchange in a Constant Number of Rounds. In *Proceedings of the 9th International Workshop on Theory and Practice in Public Key Cryptography (PKC'06)*, volume 3958 of *Lecture Notes in Computer Science*, pages 427–442. Springer, April 2006. 24, 28, 35, 36, 37

[ABR01] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *Topics in Cryptology – CT-RSA'01*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158, April 2001. 22

[AG00] N. Asokan and Philip Ginzboorg. Key-Agreement in Ad-hoc Networks. *Computer Communications*, 23(17):1627–1637, 2000. 18

[AKNRT04] Yair Amir, Yongdae Kim, Cristina Nita-Rotaru, and Gene Tsudik. On the Performance of Group Key Agreement Protocols. *ACM Transactions on Information and System Security*, 7(3):457–488, 2004. 2

[ARP03] Sattam S. Al-Riyami and Kenneth G. Paterson. Tripartite Authenticated Key Agreement Protocols from Pairings. In *Proceedings of the IMA Conference on Cryptography and Coding*, volume 2898 of *Lecture Notes in Computer Science*, pages 332–359. Springer, 2003. 8

[AST98] Giuseppe Ateniese, Michael Steiner, and Gene Tsudik. Authenticated Group Key Agreement and Friends. In *Proceedings of the 5th ACM conference on Computer and Communications Security (CCS'98)*, pages 17–26. ACM Press, 1998. 6, 14, 15, 16

[AST00] Giuseppe Ateniese, Michael Steiner, and Gene Tsudik. New Multi-Party Authentication Services and Key Agreement Protocols. *IEEE Journal of Selected Areas in Communications*, 18(4):628–639, 2000. 15

[BC04] Emmanuel Bresson and Dario Catalano. Constant Round Authenticated Group Key Agreement via Distributed Computation. In *Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography (PKC'04)*, volume 2947 of *Lecture Notes in Computer Science*, pages 115–129. Springer, 2004. 6, 28, 29, 36, 37

[BCP01] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Provably Authenticated Group Diffie-Hellman Key Exchange - The Dynamic Case. In

Advances in Cryptology – ASIACRYPT'01, volume 2248 of *Lecture Notes in Computer Science*, pages 290–390. Springer, December 2001. 5, 26, 28, 30, 31, 36, 37

[BCP02a] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions. In *Advances in Cryptology – EUROCRYPT'02*, volume 2332 of *Lecture Notes in Computer Science*, pages 321–336. Springer, Mai 2002. 5, 25, 31, 35, 36, 37

[BCP02b] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks. In *Advances in Cryptology – ASIACRYPT'02*, volume 2501 of *Lecture Notes in Computer Science*, pages 497–514. Springer, December 2002. 27, 31, 32, 36, 37

[BCP02c] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. The Group Diffie-Hellman Key Problems. In *Proceedings of the Workshop on Selected Areas in Cryptography (SAC'02)*, volume 2595 of *Lecture Notes in Computer Science*, pages 325–338. Springer, August 2002. 29, 36

[BCP06] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. A Security Solution for IEEE 802.11s Ad-hoc Mode: Password-Authentication and Group Diffie-Hellman Key Exchange. *International Journal on Wireless and Mobile Computing*, 2006. To appear. 32, 36, 37

[BCPQ01] Emmanuel Bresson, Olivier Chevassut, David Pointcheval, and Jean-Jacques Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. In *Proceedings of the 8th ACM conference on Computer and Communications Security (CCS'01)*, pages 255–264. ACM Press, 2001. 5, 29, 30, 31, 32, 34, 36, 37

[BD94] M. Burmester and Y. Desmedt. A Secure and Efficient Conference Key Distribution System. In *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 275–286. Springer, May 1994. 8, 10, 22

[BD05] M. Burmester and Y. Desmedt. A Secure and Scalable Group Key Exchange System. *Information Processing Letters*, 94(3):137–143, 2005. 10, 22

[BDS03] Rana Barua, Ratna Dutta, and Palash Sarkar. Extending Joux's Protocol to Multi Party Key Agreement. In *Progress in Cryptology – INDOCRYPT'03*,

volume 2904 of *Lecture Notes in Computer Science*, pages 205–217. Springer, December 2003. 11, 21, 22, 33

[BF03] Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003. 7, 11

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In *Advances in Cryptololgy – ASIACRYPT'01*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, December 2001. 33

[BM03] Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Springer, 2003. ISBN:3-540-43107-1. 2

[Bol03] Alexandra Boldyreva. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography (PKC'03)*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2003. 33

[Bon98] Dan Boneh. The Decision Diffie-Hellman Problem. In *ANTS-III: Proceedings of the Third International Symposium on Algorithmic Number Theory*, pages 48–63. Springer, 1998. 29

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated Key Exchange Secure Against Dictionary Attacks. In *Advances in Cryptology–EUROCRYPT'00*, volume 1807 of *Lecture Notes in Computer Science*, pages 139–155. Springer, May 2000. 25

[BR93] Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS'93)*, pages 62–73. ACM Press, 1993. 11, 24, 27, 29, 35

[BSS05] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Advances in Elliptic Curve Cryptography*. Cambridge University Press, April 2005. ISBN:0-521-60415-X. 7, 11, 14, 20

[BVS05] Jens-Matthias Bohli, Maria Isabel Gonzalez Vasco, and Rainer Steinwandt. Secure Group Key Establishment Revisited. Cryptology ePrint Archive, Report 2005/395, 2005. http://eprint.iacr.org/. 5, 6, 23, 26

[BW98] Klaus Becker and Uta Wille. Communication Complexity of Group Key Distribution. In *Proceedings of the 5th ACM Conference on Computer and Communications Security (CCS'98)*, pages 1–6. ACM Press, 1998. 17, 18

[CBH05] Kim-Kwang Raymond Choo, Colin Boyd, and Yvonne Hitchcock. Examining Indistinguishability-Based Proof Models for Key Establishment Protocols. In *Advances in Cryptology – ASIACRYPT'05*, volume 3788 of *Lecture Notes in Computer Science*, pages 585–604. Springer, 2005. 6

[CEvdG87] D. Chaum, J.-H. Evertse, and J. van de Graaf. An Improved Proof for Demonstrating Possession of Discrete Logarithms and Some Generalizations. In *Advances in Cryptology - EUROCRYPT'87*, volume 304 of *Lecture Notes in Computer Science*, pages 127–141. Springer, 1987. 10

[CFIJ99] Giovanni Di Crescenzo, Niels Ferguson, Russell Impagliazzo, and Markus Jakobsson. How to Forget a Secret. In *16th Annual Symposium on Theoretical Aspects of Computer Science (STACS'99)*, volume 1563 of *Lecture Notes in Computer Science*, pages 500–509. Springer, 1999. 25

[Che03] Zhongliang Chen. Security Analysis on Nalla-Reddy's ID-Based Tripartite Authenticated Key Agreement Protocols. Cryptology ePrint Archive, Report 2003/103, 2003. http://eprint.iacr.org/. 8

[CHL04] Kyu Young Choi, Jung Yeon Hwang, and Dong Hoon Lee. Efficient ID-based Group Key Agreement with Bilinear Maps. In *Public Key Cryptography - PKC'04*, volume 2947 of *Lecture Notes in Computer Science*, pages 130–144. Springer, March 2004. 11

[CVC04] Zhaohui Cheng, Luminita Vasiu, and Richard Comley. Pairing-Based One-Round Tripartite Key Agreement Protocols. Cryptology ePrint Archive, Report 2004/079, 2004. http://eprint.iacr.org/. 8

[DB05a] Ratna Dutta and Rana Barua. Constant Round Dynamic Group Key Agreement. In *Information Security: 8th International Conference (ISC'05)*, volume 3650 of *Lecture Notes in Computer Science*, pages 74–88. Springer, August 2005. 26, 36, 37

[DB05b] Ratna Dutta and Rana Barua. Dynamic Group Key Agreement in Tree-Based Setting. In *Proceedings of the 10th Australasian Conference on Information Security and Privacy (ACISP'05)*, volume 3574 of *Lecture Notes in Computer Science*, pages 101–112. Springer, 2005. 34, 35, 36, 37

[DB05c] Ratna Dutta and Rana Barua. Overview of Key Agreement Protocols. Cryptology ePrint Archive, Report 2005/289, 2005. http://eprint.iacr.org/2005/289/. 2

[DB06] Ratna Dutta and Rana Barua. Password-Based Encrypted Group Key Agreement. *International Journal of Network Security*, 3(1):23–34, July 2006. Available at http://isrc.nchu.edu.tw/ijns/. 27, 28

[DBS04] Ratna Dutta, Rana Barua, and Palash Sarkar. Provably Secure Authenticated Tree Based Group Key Agreement. In *Proceedings of the 6th International Conference on Information and Communications Security (ICICS'04)*, volume 3269 of *Lecture Notes in Computer Science*, pages 92–104. Springer, 2004. 33, 34, 35, 36, 37

[DH76] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976. 6, 7

[DP06] Yevgeniy Dodis and Prashant Puniya. On the Relation Between the Ideal Cipher and the Random Oracle Models. In *Third Theory of Cryptography Conference (TCC'06)*, volume 3876 of *Lecture Notes in Computer Science*, pages 184–206. Springer Verlag, 2006. 24, 35

[Gol04] Oded Goldreich. *Foundations of Cryptography - Volume II Basic Applications*, volume 2. Cambridge University Press, 2004. ISBN:0-521-83084-2. 31

[HBN04] Yvonne Hitchcock, Colin Boyd, and Juan Manuel González Nieto. Tripartite Key Exchange in the Canetti-Krawczyk Proof Model. In *Progress in Cryptology – INDOCRYPT'94*, volume 3348 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2004. 8

[ITW82] Ingemar Ingemarsson, Donald T. Tang, and C. K. Wong. A Conference Key Distribution System. *IEEE Transactions on Information Theory*, 28(5):714–719, 1982. 8, 11

[JN03] Antoine Joux and Kim Nguyen. Separating Decision DiffieHellman from Computational DiffieHellman in Cryptographic Groups. *Journal of Cryptology*, 16(4):239–247, September 2003. 11

[Jou00] Antoine Joux. A One Round Protocol for Tripartite Diffie-Hellman. In *Algorithmic Number Theory, IV-th Symposium (ANTS IV)*, volume 1838 of

Lecture Notes in Computer Science, pages 385–394. Springer, July 2000. 6, 7, 11

[KLL04] Hyun-Jeong Kim, Su-Mi Lee, and Dong Hoon Lee. Constant-Round Authenticated Group Key Exchange for Dynamic Groups. In *Advances in Cryptology – ASIACRYPT'04*, volume 3329 of *Lecture Notes in Computer Science*, pages 245–259, 2004. 25, 27, 36, 37

[KPT00] Yongdae Kim, Adrian Perrig, and Gene Tsudik. Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups. In *Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS'00)*, pages 235–244. ACM Press, 2000. 19, 21

[KPT01] Yongdae Kim, Adrian Perrig, and Gene Tsudik. Communication-Efficient Group Key Agreement. In *Proceedings of IFIP TC11 Sixteenth Annual Working Conference on Information Security (IFIP/Sec'01)*, volume 193 of *IFIP Conference Proceedings*, pages 229–244. Kluwer, 2001. 18, 19, 20

[KPT04a] Yongdae Kim, Adrian Perrig, and Gene Tsudik. Group Key Agreement Efficient in Communication. *IEEE Transactions on Computers*, 53(7):905–921, July 2004. 19, 20

[KPT04b] Yongdae Kim, Adrian Perrig, and Gene Tsudik. Tree-Based Group Key Agreement. *ACM Transactions on Information and System Security*, 7(1):60–96, February 2004. 19, 21

[Kra05] Hugo Krawczyk. HMQV: A High-Performance Secure Diffie-Hellman Protocol. In *Advances in Cryptology – CRYPTO'05*, volume 3621 of *Lecture Notes in Computer Science*, pages 546–566. Springer, 2005. 7

[KS05] Jonathan Katz and Ji Sun Shin. Modeling Insider Attacks on Group Key-Exchange Protocols. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS'05)*, pages 180–189. ACM Press, 2005. 5, 35, 37

[KY03] Jonathan Katz and Moti Yung. Scalable Protocols for Authenticated Group Key Exchange. In *Advances in Cryptology - CRYPTO'03*, volume 2729 of *Lecture Notes in Computer Science*, pages 110–125. Springer, 2003. 5, 22, 25, 26, 27, 33, 35, 36, 37

[LKKR03] Sangwon Lee, Yongdae Kim, Kwangjo Kim, and Dae-Hyun Ryu. An Efficient Tree-Based Group Key Agreement Using Bilinear Map. In *Proceedings of the*

First International Conference on Applied Cryptography and Network Security (ACNS'03), volume 2846 of *Lecture Notes in Computer Science*, pages 357–371. Springer, 2003. 21, 22

[LM06] Lijun Liao and Mark Manulis. Tree-Based Group Key Agreement Framework for Mobile Ad-Hoc Networks. In *Proceedings of 20th International Conference on Advanced Information Networking and Applications (AINA 2006)*, volume 2, pages 5–9. IEEE Computer Society, 2006. 20

[LMQ+03] Laurie Law, Alfred Menezes, Minghua Qu, Jerry Solinas, and Scott Vanstone. An Efficient Protocol for Authenticated Key Agreement. *Designs, Codes and Cryptography*, 28(2):119–134, 2003. 7

[Man05] Mark Manulis. Contributory Group Key Agreement Protocols, Revisited for Mobile Ad-Hoc Groups. In *Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'05)*, pages 811–818. IEEE Computer Society, November 2005. 11, 14, 20

[Man06] Mark Manulis. Survey on Security Requirements and Models for Group Key Exchange. Technical Report 2006/02, Horst-Görtz Institute, Network and Data Security Group, November 2006. also available at http://eprint.iacr.org/2006/388. 2, 5, 6, 34, 35, 37

[MWW98] C. J. Mitchell, Mike Ward, and Piers Wilson. Key Control in Key Agreement Protocols. *Electronic Letters*, 34(10):980–981, 1998. 6

[Nal03] Divya Nalla. ID-based Tripartite Key Agreement with Signatures. Cryptology ePrint Archive, Report 2003/144, 2003. http://eprint.iacr.org/2003/144.pdf. 8

[NK03] Divya Nalla and K.C.Reddy. ID-based Tripartite Authenticated Key Agreement Protocols from Pairings. Cryptology ePrint Archive, Report 2003/004, 2003. http://eprint.iacr.org/. 8

[Per99] A. Perrig. Efficient Collaborative Key Management Protocols for Secure Autonomous Group Communication. In *Proceedings of the International Workshop on Cryptographic Techniques and Electronic Commerce 1999*, pages 192–202. City University of Hong Kong Press, 1999. 19

[PQ01] O. Pereira and J.-J. Quisquater. A Security Analysis of the CLIQUES Protocols Suites. In *Proceedings of the 14th IEEE Computer Security Foundations*

Workshop (CSFW'01), pages 73–81. IEEE Computer Society Press, June 2001. 15

[PQ03a] O. Pereira and J.-J. Quisquater. Some Attacks upon Authenticated Group Key Agreement Protocols. *Journal of Computer Security*, 11(4):555–580, 2003. 15

[PQ03b] Olivier Pereira and Jean-Jacques Quisquater. An Attack against Barua *et al.* Authenticated Group Key Agreement Protocol. Technical Report CG-2003-3, UCL Crypto Group, October 2003. 22

[RH03] Sandro Rafaeli and David Hutchison. A Survey of Key Management for Secure Group Communication. *ACM Computer Surveys*, 35(3):309–329, 2003. 2

[Sch89] Claus P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989. 32

[SD] J. Schwenk and Deutsche Telekom AG. Deutsches Patent DE19847941. 20

[SH03] Hung-Min Sun and Bin-Tsan Hsieh. Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings. Cryptology ePrint Archive, Report 2003/113, 2003. http://eprint.iacr.org/2003/113. 8

[Sha49] C. E. Shannon. Communication Theory of Secrecy Systems. *The Bell Systems Technical Journal*, 28(4):656–715, 1949. 24, 35

[Shi03a] K. Shim. Efficient One-Round Tripartite Authenticated Key Agreement Protocol from the Weil Pairing. *Electronics Letters*, 39(2):208–209, January 2003. 8

[Shi03b] Kyungah Shim. Cryptanalysis of Al-Riyami-Paterson's Authenticated Three Party Key Agreement Protocols. Cryptology ePrint Archive, Report 2003/122, 2003. http://eprint.iacr.org/. 8

[Shi03c] Kyungah Shim. Cryptanalysis of ID-based Tripartite Authenticated Key Agreement Protocols. Cryptology ePrint Archive, Report 2003/115, 2003. http://eprint.iacr.org/. 8

[Sho99] Victor Shoup. On Formal Models for Secure Key Exchange (Version 4). Technical Report RZ 3120, IBM Research, November 1999. Also available at http://shoup.net/. 5, 25

[SMS01] J. Schwenk, T. Martin, and R. Schaffelhofer. Tree-based Key Agreement for Multicast. In *Proceedings of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security Issues*, volume 192 of *IFIP Conference Proceedings*. Kluwer, 2001. 20

[SSDW90] D. G. Steer, L. Strawczynski, Whitfield Diffie, and Michael J. Wiener. A Secure Audio Teleconference System. In *Advances in Cryptology – CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 520–528. Springer, 1990. 8, 16, 17

[Ste02] Michael Steiner. *Secure Group Key Agreement*. PhD thesis, Saarland University, March 2002. 5

[STW96] Michael Steiner, Gene Tsudik, and Michael Waidner. Diffie-Hellman Key Distribution Extended to Group Communication. In *Proceedings of the 3rd ACM Conference on Computer and Communications Security (CCS'96)*, pages 31–37. ACM Press, 1996. 12, 13, 14, 18, 29

[STW98] Michael Steiner, Gene Tsudik, and Michael Waidner. CLIQUES: A New Approach to Group Key Agreement. In *Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS'98)*, pages 380–387. IEEE Computer Society Press, 1998. 13, 14, 15

[STW00] Michael Steiner, Gene Tsudik, and Michael Waidner. Key Agreement in Dynamic Peer Groups. *IEEE Transactions on Parallel and Distributed Systems*, 11(8):769–780, 2000. 14

[ZC03] Fangguo Zhang and Xiaofeng Chen. Attack on Two ID-based Authenticated Group Key Agreement Schemes. Cryptology ePrint Archive, Report 2003/259, 2003. Available at http://eprint.iacr.org/2003/259/. 11

[ZLK02] Fangguo Zhang, Shengli Liu, and Kwangjo Kim. ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings. Cryptology ePrint Archive, Report 2002/122, 2002. http://eprint.iacr.org/2002/122. 8, 22