A Secure and Scalable Group Key Exchange System*

Mike Burmester, Yvo Desmedt†

Computer Science Department, Florida State University, Tallahassee, FL 32306-4530, USA

Abstract

We present a Group Key Exchange protocol which extends in a natural way the Diffie-Hellman protocol. Our protocol is scalable: it has two rounds (for $n > 2$ parties) and the number of modular exponentiations per user is constant. It is secure against a passive adversary if the Diffie-Hellman problem is intractable.

Keywords: Group Key Exchange, Diffie-Hellman, cryptography.

1 Introduction

To communicate securely over an insecure public network messages must be encrypted. Key Exchange (KE) protocols allow two parties to exchange a secret encryption key. With Group Key Exchange (Conference Key Distribution) protocols, groups of two or more parties exchange a *group* secret key. KE is central to cryptography and has attracted a lot of attention (e.g., [13, 17]). Research has focused on both security and efficiency. Many practical systems have been proposed in the literature (e.g., [14, 32]). The most familiar KE protocol is the Diffie-Hellman protocol [13]. Several Group Key Exchange (GKE) protocols have also been proposed. Some of the earlier ones [17, 22, 15, 6] had inadequate security or were rather impractical. In general, designing GKE protocols can be particularly challenging because of the complexity of the interactions between the many parties. Several provably secure Authenticated GKE protocols have been proposed [11, 12, 21, 8]. These protocols improve significantly on earlier work, however they are not scalable: either the number of rounds required is $O(n)$, or the number of exponentiations is $O(n)$ (n is the number of group members).

In this paper we present a scalable GKE protocol which is proven secure in the passive adversary case, with forward secrecy [14], if the Diffie-Hellman problem is intractable. Our protocol extends in a natural

*The protocol in this paper was presented at Eurocrypt '94 and appeared in the Proceedings [10]. The proof of security for groups with an even number of parties appeared in the Pre-proceedings [9].

†After submission of this paper, Yvo Desmedt has moved to University College, London, UK.

way the Diffie-Hellman protocol to $n > 2$ parties and is based on one of the Ingemarsson, Tang and Wong protocols [17] which uses symmetric functions. That protocol has many attractive features, however it is insecure because the information exchanged by the group members makes it possible for a passive adversary to compute the key [17]. Our protocol uses a *cyclic* function of degree 2, which prevents this attack while retaining the efficiency of the former protocol. It has two rounds and requires $O(1)$ (constant) modular exponentiations per user. We do not address group robustness –see [29, 21, 27] for the dynamics of groups.

Research and subsequent work. The motivation for this paper is to give a security proof for the protocol first presented at Eurocrypt 94 [10]. Since this protocol first appeared, it has been used extensively in the literature because of its efficiency [24, 26]. However, several authors have commented on the lack of proof of security. Indeed since no security proofs were given in [10], subsequent work (e.g., [12, 8]) viewed this protocol as being purely heuristic. Bresson-Chevassut-Pointcheval [11] and Bresson-Chevassut-Pointcheval-Quisquater [12] use the GKE protocol of Steiner-Tzudik-Weidner [28] that requires $O(n)$-rounds, while our protocol requires only 2 rounds. Recently, Katz-Yung [20] analyzed GKE protocols in a formal security model based on indistinguishability and described a scalable compiler that transforms *any* GKE that is secure against a passive adversary into an authenticated GKE that is secure against an adversary who controls all communication in the network. They proved that in this model the security of our protocol reduces to the Decisional Diffie-Hellman (DDH) problem. This problem is regarded as intractable for most settings. However, in the Gap Diffie-Hellman group it can be solved in polynomial time, while no probabilistic algorithm can solve the Computational Diffie-Hellman problem with non-negligible advantage in polynomial time [7, 18, 19].

2 Definitions

We consider public networks in which members of a group can *broadcast* messages (bit strings) to each other in the presence of a (polynomially bounded) passive adversary $\mathcal{A}$ (an eavesdropper). $\mathcal{A}$ may read the broadcast messages and keep a log of transcripts of past executions (a history) but cannot modify messages.

Definition 2.1. Let k be a security parameter and $U_1, U_2, \ldots, U_n$, $n = poly(k)$ (a polynomial in k), be members (interactive polynomial-time Turing Machines with history tapes) of a group that take part in a protocol $\mathcal{P}$ to generate a group key. Protocol $\mathcal{P}$ is called a *Group Key Exchange* (GKE) if: when all group members follow the protocol as specified, then each member U_i will compute the same key $K = K_i$.

Definition 2.2. Let $\mathcal{P}$ be a GKE protocol and $\mathcal{A}$ a passive adversary. Assume that $\mathcal{A}$ has witnessed polynomially-many instances of $\mathcal{P}$ and let K be the key output by the last instance.

1. $\mathcal{P}$ guarantees *privacy* if it is computationally infeasible for $\mathcal{A}$ to compute K.

2. $\mathcal{P}$ guarantees *secrecy* if $\mathcal{A}$ cannot distinguish K from a random bit string of the same length with probability better than $\frac{1}{2} + \varepsilon$, where ε is negligible (in k).

Remark 2.3. Privacy was used in [9] to show that our GKE protocol is a natural multi-party extension of the original (two party) Diffie-Hellman protocol. Secrecy is a stronger version of security, and is currently used as the definition of security for passive adversaries. It was introduced for KE protocols by Bellare-Rogaway [1] (and Bird et al. [5]), and later extended for multi-party applications by Bresson et al. [11, 12]. To convert a requirement that it is hard to compute a key into a requirement that the key is pseudorandom, one can use the random-oracle heuristic of Bellare-Rogaway [1] in which keys are randomized by using a function selected randomly from a family of universal hash functions. This converts a computational (or search) problem into a decisional problem. In applications, one may use the the Secure Hash Algorithm SHA-1 [26, 24].

Definition 2.4. *Let $Z_p = Z/pZ$ be the integers modulo p, and let $g \in Z_p$ and $G = \{g^x \bmod p, x \in Z\}$ the integers modulo p generated by g. The following problems refer to computations or decisions in G.*

1. The Computational Diffie-Hellman (CDH) problem [13] (a search problem): given numbers p, g and random numbers $x, y \in G$, find $x^{\log_g y} \bmod p$. That is, if $x = g^a \bmod p$ and $y = g^b \bmod p$, find $g^{ab} \bmod p$.

2. The Decisional Diffie-Hellman (DDH) problem [13]: given numbers p, g and random numbers $x, y, z \in G$, decide if: $z = x^{\log_g y} \bmod p$.

3. The squaring Computational Diffie-Hellman (s-CDH) problem [23, 30]: given numbers p, g and a random number $x \in G$, find $x^{\log_g x} \bmod p$. That is, if $x = g^a \bmod p$, find $g^{a^2} \bmod p$.

4. The squaring Decisional Diffie-Hellman (s-DDH) problem [31]: given numbers p, g and random numbers $x, z \in G$, decide if: $z = x^{\log_g x} \bmod p$.

The generator g is usually taken to be a unit of Z_p, in which case the problems involve computations or decisions in cyclic subgroups (G, g) of the group Z_p^* of units of Z_p. These problems extend in a natural way to problems in general families of cyclic groups $\{(G, g)\}$ generated by sampler functions, whose group

operation can be computed efficiently (in k).[1] For most applications p is prime, and sometimes the order of g is also prime. In this paper we shall not restrict ourselves to groups of prime order. However it is important that the group parameters be selected appropriately, if we want the problem to be intractable (for example, the order of g must have a large prime factor; also for the s-DDH and DDH problems, if p is prime then g must be a quadratic residue modulo p). For appropriate group parameters, both the CDH and the DDH problem have remained intractable for more than 25 years. However, in the Gap Diffie-Hellman group the DDH problem is tractable, while the CDH problem is intractable [7, 18, 19].

The equivalence of s-CDH and CDH and the equivalence of s-DDH and DDH respectively, have been discussed and researched extensively in the literature (see *e.g.*, [23, 31, 30]). For the computational problems we have equivalence in the generic case (with the same complexity) under the high and medium granularity assumptions [23, 30], but for low granularity we do not know. For the decisional problems we know less [23, 30]. However we do know that there is no reduction from DDH to s-DDH in the generic model [23, 31].

3 The Group Key Exchange protocol

A center chooses the parameters of the system: a security parameter k, a prime $p = \Theta(2^{ck})$, $c \geq 1$ constant, and an element $g \in Z_p$ of order $q = \Theta(2^k)$. If this has to be verified then the factorization of q is given (alternatively, a zero-knowledge proof [16] that g has order q is given). The center publishes p, g and q.

Protocol 1. Let $U_1, U_2, \ldots, U_n$, $n = poly(k)$, be a group of parties[2] that want to generate a group key.

Round 1. Each party U_i, $i = 1, 2, \ldots, n$, selects a random $r_i \in Z_q$, and broadcasts[3] $z_i := g^{r_i} \bmod p$.

Round 2. Each party U_i, $i = 1, 2, \ldots, n$, broadcasts $X_i := (z_{i+1}/z_{i-1})^{r_i} \bmod p$, where the indices are taken in a cycle.

Key Computation. Each party U_i, $i = 1, 2, \ldots, n$, computes the key:

$$K_i := (z_{i-1})^{nr_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i-2} \bmod p. \tag{1}$$

Lemma 3.1 Protocol 1 is a Group Key Exchange. That is, if all parties adhere to the protocol then each will compute the same key:

$$K = g^{r_1 r_2 + r_2 r_3 + \cdots + r_n r_1} \bmod p.$$

[1] For a general classification of discrete-log based problems see Steiner [27].
[2] The group is dynamic: our only restriction is that it is polynomially bounded.
[3] It suffices to send it to U_{i+1}.

Indeed, let $A_{i-1} \equiv (z_{i-1})^{r_i} \equiv g^{r_{i-1}r_i} \,(\mathrm{mod} p)$, $A_i \equiv (z_{i-1})^{r_i} \cdot X_i \equiv g^{r_i r_{i+1}} \,(\mathrm{mod} p)$, $A_{i+1} \equiv (z_{i-1})^{r_i} \cdot X_i \cdot X_{i+1} \equiv g^{r_{i+1}r_{i+2}} \,(\mathrm{mod} p)$, etc. Then $K_i \equiv A_{i-1} \cdot A_i \cdot A_{i+1} \cdots A_{i-2} \equiv K \,(\mathrm{mod} p)$. □

In the special case when there are only two parties, $X_1 = X_2 = 1$ and $K \equiv g^{r_1 r_2 + r_2 r_1} \equiv g^{2r_1 r_2} (\mathrm{mod} p)$. This is essentially the Diffie-Hellman KE (in this case there is no need to broadcast X_1, X_2).

Theorem 3.2. [10] *Let n be even. Protocol 1 guarantees privacy if, and only if, the CDH problem is intractable.*

Proof. [9] If the CDH problem is feasible then clearly the group key K can easily be computed from the broadcast data. Suppose that the CDH problem is intractable. We shall prove that Protocol 1 guarantees privacy by contradiction. Let $\mathcal{A}$ be a passive adversary that succeeds with non-negligible probability in computing the group key $K_1 = (z_n)^{nr_1} \cdot X_1^{n-1} \cdot X_2^{n-2} \cdots X_{n-1} \bmod p$ of U_1. We show how to use the program of $\mathcal{A}$ to solve the CDH problem by simulating its input. First observe that it is easy to simulate a history of transcripts of previous executions by selecting for each execution the exponents $r_i \in Z_q$, $i = 1, 2, \ldots, n$ at random.

Let $p, g, z_n, z_1,$ be an instance of the CDH problem. Input to $\mathcal{A}$: p, g, a history of transcripts and the transcript $z_1, z_2, \ldots, z_n$ and $X_1, X_2, \ldots, X_n$ obtained from the CDH instance as follows. Select $b_2, b_3, \ldots, b_{n-1}$ at random from Z_q and compute:

$$z_2 := z_n \cdot g^{b_2} \bmod p,\ z_3 := z_1 \cdot g^{b_3} \bmod p,\ z_4 := z_2 \cdot g^{b_4} \bmod p,\ \ldots\ ,\ z_{n-1} := z_{n-3} \cdot g^{b_{n-1}} \bmod p. \quad (2)$$

Then it is easy to compute $X_1 :\equiv {z_1}^{b_2} \equiv (g^{\log_g z_1})^{b_2} \equiv (g^{b_2})^{\log_g z_1} \equiv (z_2/z_n)^{\log_g z_1} \,(\mathrm{mod} p)$, and similarly $X_2, X_3, \ldots, X_{n-2}$. From (2), since n is even, we get,

$$\begin{array}{rcccccccl} z_n & \equiv & z_2 \cdot g^{-b_2} & \equiv & z_4 \cdot g^{-b_2-b_4} & \equiv & \ldots & \equiv & z_{n-2} \cdot g^{-b_2-b_4-\ldots-b_{n-2}} \,(\mathrm{mod} p), \\ z_1 & \equiv & z_3 \cdot g^{-b_3} & \equiv & z_5 \cdot g^{-b_3-b_5} & \equiv & \ldots & \equiv & z_{n-1} \cdot g^{-b_3-b_5-\ldots-b_{n-1}} \,(\mathrm{mod} p). \end{array} \quad (3)$$

So one can compute: $X_{n-1} :\equiv (z_{n-1})^{-b_2-b_4-\ldots-b_{n-2}} \,(\mathrm{mod} p)$. Indeed, from (3) and the definition of X_{n-1}, we have: $X_{n-1} \equiv (z_n/z_{n-2})^{\log_g z_{n-1}} \equiv g^{(-b_2-b_4-\ldots-b_{n-2})*\log_g z_{n-1}} \equiv (z_{n-1})^{-b_2-b_4-\ldots-b_{n-2}} \,(\mathrm{mod} p)$. Similarly $X_n :\equiv (z_n)^{-b_3-b_5-\ldots-b_{n-1}} \equiv (z_1/z_{n-1})^{\log_g z_n} \,(\mathrm{mod} p)$. Note that the probability distribution of $z_2, z_3, \ldots, z_{n-1}$ and $X_1, X_2, \ldots, X_n$ is *identical* to that of the real input of the adversary $\mathcal{A}$.

We now have the necessary input for $\mathcal{A}$: p, g, a history, and the transcript $z_1, \ldots, z_n$, $X_1, \ldots, X_n$. By our assumption $\mathcal{A}$ will output the corresponding key K_1. From K_1, by using (1) with $i = 1$, it is easy to compute $(z_n)^{n \log_g z_1} \bmod p$, since all the X_i are known. It is well known [4, 25] that it is feasible to compute

n-th residues in Z_p when n is polynomially bounded. It may not be possible to find the exact residue that corresponds to $(z_n)^{\log_g z_1} \bmod p$, if there are many (if q is prime there is only one), so choose one at random. With non-negligible probability, bounded by $1/n$, this will be the right one. Consequently, by using the program of $\mathcal{A}$ we get $(z_n)^{\log_g z_1} \bmod p$ with non-negligible probability. Since no restrictions are put on the input p, g, z_n, z_1, we get a solution for a general instance of the CDH problem. □

Corollary 3.3. [10] *Theorem 3.2 can easily be extended to allow for the case when the number of parties is odd by slightly modifying Protocol 1.*

Proof. [9] If the number of parties is odd then one party, say the last one, behaves virtually as two independent machines. Alternatively, for symmetry, all parties behave as two virtual machines. □

In the next two theorems we will not need to distinguish the cases when the number of parties is even or odd.

Theorem 3.4. *Protocol 1 guarantees secrecy if, and only if, the DDH problem is intractable.*

Proof. (See also [20]) This is similar to the proof of Theorem 3.2. Let $\mathcal{A}$ be a passive adversary that can distinguish the key K_1 from a random key with probability better than guessing, and let p, g, z_n, z_1, z be an instance of the DDH problem. Input to $\mathcal{A}$: p, g, a history and the transcript $z_1, z_2, \ldots, z_n$ and $X_1, X_2, \ldots, X_n$ obtained from the DDH instance as follows. Select $b_2, b_3, \ldots, b_{n-1}$ at random from Z_q and compute:

$$z_i := g^{b_i} \bmod p, \;\; X_i :\equiv (z_{i+1}/z_{i-1})^{b_i} \equiv (z_{i+1}/z_{i-1})^{\log_g z_i} \;(\mathrm{mod}p), \quad i = 2, 3, \ldots, n-1,$$

and $X_1 :\equiv z_1^{b_2}/z \equiv (z_2)^{\log_g z_1}/z \;(\mathrm{mod}p)$, $X_n :\equiv z/(z_n)^{b_{n-1}} \equiv z/(z_{n-1})^{\log_g z_n} \;(\mathrm{mod}p)$. Note that in X_1 and X_n we have replaced the Diffie-Hellman value $(z_n)^{\log_g z_1} \equiv (z_1)^{\log_g z_n} \;(\mathrm{mod}p)$ by the "test" value z. By our assumption, $\mathcal{A}$ will succeed for the given instance of the DDH problem. The converse is trivial. □

Theorem 3.5. *Protocol 1 guarantees privacy (secrecy) if the s-CDH problem (the s-DDH problem) is intractable.*

Proof. First we deal with privacy. Let $\mathcal{A}$ be a passive adversary that can compute the key $K = K_1$ with non-negligible probability and let $p, g, z_n \in Z_p$ be an instance of the s-CDH problem. Input to $\mathcal{A}$: p, g, a history and the transcript $z_1, z_2, \ldots, z_n$, $X_1, X_2, \ldots, X_n$ obtained from the s-CDH instance as follows. Select $b_1, b_2, \ldots, b_{n-1}$ at random from Z_q and compute:

$$z_1 := z_n \cdot g^{b_1} \bmod p, \;\; z_2 := z_n \cdot g^{b_2} \bmod p, \;\; \ldots \;\; , \;\; z_{n-2} := z_n \cdot g^{b_{n-2}} \bmod p, \;\; z_{n-1} := z_n \cdot g^{b_{n-1}} \bmod p,$$

and $X_1 :\equiv z_1^{b_2} \equiv (g^{b_2})^{\log_g z_1} \equiv (z_2/z_n)^{\log_g z_1} (\mathrm{mod} p)$, $X_2 :\equiv z_2^{b_3 - b_1} \equiv (z_3/z_1)^{\log_g z_2} (\mathrm{mod} p)$, $\ldots$, $X_{n-1} :\equiv z_{n-1}^{-b_{n-2}} \equiv (z_n/z_{n-2})^{\log_g z_{n-1}} (\mathrm{mod} p)$ and $X_n :\equiv z_n^{b_1 - b_{n-1}} \equiv (z_1/z_{n-1})^{\log_g z_n} (\mathrm{mod} p)$. So we get the input for $\mathcal{A}$. Let K_1 be the output. From K_1 we get $(z_n)^{\log_g z_1} \mathrm{mod} p$ using (1) and as in last lines of the proof of Theorem 3.2. Then compute:

$$(z_n)^{\log_g z_1} \cdot z_n^{-b_1} \equiv (z_1)^{\log_g z_n} \cdot (g^{-b_1})^{\log_g z_n} \equiv (z_1 \cdot g^{-b_1})^{\log_g z_n} \equiv (z_n)^{\log_g z_n} \pmod{p}. \tag{4}$$

This is a solution for the given instance of the s-CDH problem.

Next consider secrecy. Suppose that $\mathcal{A}$ can distinguish K_1 from a random key, and let p, g, z_n, z be an instance of the s-DDH problem. Input to $\mathcal{A}$: p, g, a history and the transcript $z_1, z_2, \ldots, z_n$, $X_1, X_2, \ldots, X_n$ obtained from the s-DDH instance by selecting $b_1, b_2, \ldots, b_{n-1}$ at random from Z_q and computing,

$$z_1 := z_n \cdot g^{b_1} \bmod p,\ z_2 := g^{b_2} \bmod p,\ z_3 := g^{b_3} \bmod p,\ \ldots\ ,\ z_{n-1} := g^{b_{n-1}} \bmod p.$$

Then $X_i := (z_{i+1}/z_{i-1})^{b_i} \mathrm{mod} p$, for $i = 2, 3, \ldots, n-1$. Replace the s-CDH value $(z_n)^{\log_g z_n} \bmod p$ in (4) by the "test" value z to get $(z_n)^{\log_g z_1} \cdot z_n^{-b_1} \equiv z \pmod{p}$ and thus $z \cdot z_n^{b_1} \equiv (z_n)^{\log_g z_1} \pmod{p}$. Then $X_1 :\equiv z_1^{b_2}/(z \cdot z_n^{b_1}) \equiv (z_2/z_n)^{\log_g z_1} \pmod{p}$. Similarly $X_n :\equiv (z \cdot z_n^{b_1})/z_n^{b_{n-1}} \equiv (z_1/z_{n-1})^{\log_g z_n} \pmod{p}$, since $(z_n)^{\log_g z_1} \equiv (z_1)^{\log_g z_n} \pmod{p}$. So we get the input for $\mathcal{A}$. By our assumption, $\mathcal{A}$ will succeed for the given instance of the s-DDH problem. □

Corollary 3.6. *Theorems 3.2 and 3.5 can easily be extended to any family of groups (G, g) for which the order of g is a known prime q.*

Proof. The proofs are identical, except that there is no need to compute n-th residue roots. Instead $(z_n)^{n \log_g z_1} \bmod p$ (for Theorem 3.2) and $(z_n)^{n \log_g z_n} \bmod p$ (for Theorem 3.5), are raised to the power $n^{-1} \bmod q$. □

Remark 3.7. Protocol 1 focuses on security in the passive adversary case. For Authenticated GKE we may use the Katz-Yung compiler [20] mentioned earlier, which will transform a GKE that is secure against a passive adversary into an authenticated GKE that is secure against an active adversary.

Remark 3.8. We conclude by considering the possible links between Theorem 3.2, Corollary 3.3, Theorem 3.4 and Theorem 3.5. Let $A \Rightarrow B$ denote that problem A is reducible[4] to problem B. We have:

[4] A is reducible to B if there is an algorithm that transforms any solution of B to a solution of A, that is: if A is hard then so is B.

1. If s-CDH $\Rightarrow$ CDH then Theorem 3.2 and Corollary 3.3 imply Theorem 3.5 (privacy).

2. If s-DDH $\Rightarrow$ DDH then Theorem 3.4 implies Theorem 3.5 (secrecy).

3. If CDH $\Rightarrow$ s-CDH then Theorem 3.5 (privacy) implies Theorem 3.2 and Corollary 3.3.

4. If DDH $\Rightarrow$ s-DDH then Theorem 3.5 (secrecy) implies Theorem 3.4.

From [30] (which extends the results in [23]) we have the following relationships between the problems CDH, DDH, s-CDH and s-DDH:

1. It is unlikely that DDH $\Rightarrow$ s-DDH, because this does not hold in the generic model.

2. CDH $\Rightarrow$ s-CDH for high and medium granularity, but it is not known if this extends to low granularity. The same holds for: s-CDH $\Rightarrow$ CDH and s-DDH $\Rightarrow$ DDH (although there are indications that the last two reductions may hold for low granularities). These reductions apply only to subgroups of *known order*.

Acknowledgments

The authors wish to thank Tom Berson, Kevin McCurley, Oded Goldreich, René Peralta, Paul van Oorschot, Adi Shamir and Moti Yung for helpful discussions. We also wish too thank Stanislaw Jarecki for his support and Remark 3.8.

References

[1] M. Bellare and P. Rogaway. Entity Authentication and Key Distribution. *Crypto '93, LNCS 773*, Springer-Verlag, Berlin, 232–249, 1994

[2] Mihir Bellare, Phillip Rogaway. Provably Secure Session Key Distribution: the Three Party Case. STOC 1995: 57-66, 1995

[3] M. Bellare, D. Pointcheval and P. Rogaway. Authenticated Key Exchange Secure Against Dictionary Attacks. *Eurocrypt '2000, LNCS 1807*, Springer-Verlag, Berlin, 139–155, 2000

[4] E. R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, **24**(111): 713–735, 1970

[5] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva and M.Yung. Systematic design of two-party authentication protocols. *Crypto '91, LNCS 576*, Springer-Verlag, Berlin, 44–61, 1992

[6] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. *Proc. Crypto '92, LNCS 740*, Springer-Verlag, Berlin, 471–486, 1993

[7] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *Proc. Crypto '2001, LNCS 2139*, Springer-Verlag, Berlin, 213–229, 2001.

[8] C. Boyd, J.M.G. Nieto. Round-Optimal Contributory Conference Key Agreement. *PKC 2003, LNCS 2567*, Springer-Verlag, Berlin, 161-174, 2003

[9] M. Burmester and Y Desmedt. A Secure and Efficient Conference Key Distribution System. *Preproceedings of Eurocrypt '94*, Scuola Superiore Guglielmo Reiss Romoli (SSGRR), pp. 279–290, Perugia, Italy, May 9–12, 1994

[10] M. Burmester and Y Desmedt. A Secure and Efficient Conference Key Distribution System. *Eurocrypt '94, LNCS 950*, Springer-Verlag, Berlin, 275–286, 1995

[11] E. Bresson, O. Chevassut and D. Pointcheval. Group Diffie-Hellman Key Exchange under Standard Assumptions. *Eurocrypt 2001, LNCS 2045*, Springer-Verlag, Berlin, 321–336, 2002

[12] E. Bresson, O. Chevassut, D. Pointcheval and J.-J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. *ACM Conf. Comp. & Comm. Security*: 255-264, 2001

[13] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT–22(6): 644–654, 1976

[14] P. C. van Oorschot W. Diffie and M. J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2: 107–125, 1992

[15] M. J. Fischer and R. N. Wright. Multiparty secret key exchange using a random deal of cards. *Crypto '91, LNCS576*, Springer-Verlag, Berlin, 141–155, 1992

[16] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1): 186–208, 1989

[17] I. Ingemarsson, D. T. Tang, and C. K. Wong. A conference key distribution system. *IEEE Trans. Inform. Theory*, 28(5): 714–720, 1982

[18] A. Joux. A one round protocol for tripartite Diffie–Hellman. *Proc. ANTS-4, LNCS 1838*, Springer-Verlag, Berlin, 385–393, 2000.

[19] A. Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. *Proc. ANTS-5, LNCS 2369*, Springer-Verlag, Berlin, 20–32, 2002.

[20] J. Katz, and M. Yung. Scalable Protocols for Authenticated Group Key Exchange. *Crypto 2003, LNCS 2729*, Springer-Verlag, Berlin, 110-125, 2003.

[21] Y. Kim, A. Perrig and G. Tsudik. Communication-Efficient Group Key Agreement, *Proc. IFIP SEC 2001*, June 2001.

[22] K. Koyama and K. Ohta. Identity-based conference key distribution systems. *Crypto '87, LNCS 293*, Springer-Verlag, Berlin, 175–185, 1988

[23] U. Maurer and S. Wolf. Diffie-Hellman Oracles. *Crypto '96, LNCS 1109*, Springer-Verlag, Berlin, 268–282, 1996

[24] A. Menezes, P.C. van Oorschot and S.A. Vanstone. Handbook of Applied Cryptography, CRC Press, 1997.

[25] M. Rabin. Probabilistic algorithms in finite fields. *SIAM J. on Computing*, 9(2): 273–280, 1980

[26] B. Schneier, Applied Cryptography, John Wiley & Sons, 1996.

[27] M. Steiner. Secure Group Key Agreement. *Naturwissenschaftlich-Technische Facultät der Universität des Saarlandes, Saarbrücken*, Dissertation, March 2002
http://www.semper.org/sirene/publ/Stei_02.thesis-final.pdf

[28] M. Steiner, G. Tsudik and M. Waidner. Diffie-Hellman Key Distribution Extended to Group Communication. *ACM Conf. Comp. & Comm. Security*: 31–37, 1996

[29] M. Steiner, G. Tsudik and M. Waidner. Key Agreement in Dynamic Peer Groups. *IEEE Trans. Parallel and Distributed Systems*, **11**(8), pp. 769–780, 2000.

[30] A-R. Sadeghi and M. Steiner. Assumptions Related to Discrete Logarithms: Why Subtleties Make a Real Difference. *Eurocrypt 2001, LNCS 2045*, Springer, pp. 244–260, 2001

[31] S. Wolf. Information-theoretically and Computationally Secure Key Agreement in Cryptography. Ph.D. Thesis, ETH Zurich, 1999.

[32] Y. Yacobi and Z. Shmuely. On key distribution systems. *Crypto '89, LNCS 435*, Springer-Verlag, Berlin, pp. 344–355, 1990