Communication Complexity of Group Key Distribution

Klaus Becker
r^3 security engineering ag
CH–8301 Zürich

Uta Wille
IBM Research Division
Zürich Research Laboratory
CH–8803 Rüschlikon

Abstract

Communication complexity has always been an important issue when designing group key distribution systems. This paper systematically studies what can be achieved for the most common measures of protocol complexity. Lower bounds for the total number of messages, the total number of exchanges, and the number of necessary rounds are established, whereby models that allow broadcasting have to be distinguished from those that do not. For every measure of protocol complexity, we furthermore show that the corresponding bound is realistic for Diffie–Hellman-based protocols by referring to or introducing protocols that match the bound or exceed it by only one.

1 Introduction

Since the publication of 2-party Diffie–Hellman (DH) key exchange in 1976, various solutions have been proposed to extend Diffie–Hellman key exchange to multiparty key distribution. Most notable and best known among those proposals are the protocols by Ingemarson et al. [ITW82] and Burmester and Desmedt [BD94]. Beyond the security of the systems, protocol complexity has always been an important issue when designing group key distribution systems. Steiner et al. [STW96], for instance, defined a class of "generic n-party DH protocols" for which they showed that security is based on the intractability of the Diffie–Hellman problem. Subsequently, they introduced two protocols that have proved to be optimal within the class with respect to certain measures of protocol complexity. Following this line of research, we systematically analyze Diffie–Hellman-based key distribution protocols in terms of protocol complexity (see [Be97]). Lower bounds for the total number of messages, the total number of exchanges, and the number of necessary rounds are established, whereby models that allow broadcasting have to be distinguished from those that do not. For every measure of protocol complexity, we show that the corresponding bound is realistic for DH-based protocols by referring to or introducing protocols that match the bound or exceed it by only one.

The overall objective of multiparty key distribution systems is to establish securely, for a distinguished set of users, a common secret communication key. If the group key is generated and distributed by a central trusted party, then it is not necessary to discuss the communication complexity. Therefore, we are concerned with the efficiency of **contributory** group key distribution systems, which interactively establish a group key such that

- no user is required to hold secret information before entering the protocol,
- each group member makes an independent contribution to the group key.

The 2-party Diffie–Hellman key exchange is a simple example of a contributory group key distribution system. We call a contributory key distribution system **Diffie–Hellman-based** if its security relies upon the intractability of the Diffie–Hellman decision problem; i.e., if the resulting key is always polynomially indistinguishable from a random number, presuming the same is true for 2-party DH keys. This concept is more general than the Diffie–Hellman-based "generic n-party DH protocols" introduced by Steiner et al. [STW96]. They require the keys to be of the form $K = \alpha^{N_1 \cdot \dots \cdot N_n}$, where $N_1, \dots, N_n$ are the random values chosen by the individual group members, and assume that all values $\alpha^{\prod_{i \in I} N_i}$ with $I \subseteq \{1, \dots, n\}$ might be revealed by the protocol.

When studying lower bounds for the communication complexity of contributory key distribution systems, we distinguish between messages, exchanges, and broadcasts. A **message** is a single package of information sent from one member to a single other member of the group. In an **exchange**, on the other hand, two parties may simultaneously exchange a message; i.e., a member M_1 is allowed to send a message to party M_2, who can simultaneously send a message to M_1. A **broadcast** is a message sent by a user and received by every other member of the group. In order to measure time complexity, we assume a synchronous round model, which presumes synchronized clocks ticking at discrete instances, where each tick is considered to be a new **round**. With respect to the design of communication efficient protocols, the following measures of protocol complexity are the ones most commonly found in the literature:

- **total messages:** total number of messages plus total number of broadcasts sent according to the protocol.
- **total exchanges:** total number of exchanges plus total number of broadcasts necessary in order to perform the protocol.

5th Conference on Computer & Communications Security
San Francisco CA USA
Copyright ACM 1998 1-58113-007-4/98/11...$5.00

- **synchronous rounds:** minimum number of synchronous rounds required by the protocol, presuming that every party is allowed to send arbitrarily many messages with every time tick and to receive arbitrarily many messages sent by other parties at the beginning of a round (cf. [BM93] p. 133).
- **simple rounds:** minimum number of required synchronous rounds, presuming that every party sends and receives at most one message per round (cf. [ABM87]).

Whereas the total number of messages of a protocol measures the number of packages sent, the number of exchanges can be considered to be the number of connections to be established during a protocol run. It should be mentioned that, when counting a broadcast as a single message sent, we implicitly assume that the underlying network is completely connected. Finally, the number of rounds is obviously an abstract measure of the time needed to perform the protocol, where simple rounds are usually considered if sending and receiving messages incurs high costs relative to the transmission of data, for instance owing to software delays encountered when moving through the protocol layers. Furthermore, for the formulation of lower bounds for the number of rounds, the distinction between synchronous and simple rounds is only relevant if broadcasting is allowed. This is indeed the case because recording synchronous rounds essentially means to permit broadcasts, i.e., different messages can be sent to different parties in one combined broadcast. In the following section, we will prove lower bounds for the aforementioned measures of protocol complexity and then discuss various protocols that match the bounds formulated in Section 3.

2 Bounds for Contributory Key Distribution Systems

Contributory key distribution protocols interactively generate a common group key including an independent contribution of every group member. With other words, n parties each contribute a secret random value to a common key, which has to be known by every party after the protocol has finished. Considering only the information that has to be distributed in order to establish a contributory group key, we observe that each member of the group has a piece of information, which has to reach every other member of the group. Therefore, the minimum number of exchanges or messages needed to distribute this information may serve as a lower bound for the number of exchanges or messages required by a contributory key distribution.

The question concerning how many telephone calls are needed to distribute n pieces of information held by n different parties to all the participating parties is known in the literature as the *gossip problem*, see [BS72] and [Be72]. Baker and Shostak [BS72] have shown that the minimum number of required phone calls is $2 \cdot n - 4$, assuming that $n \geq 4$. This result immediately provides us with a lower bound for the total number of exchanges required by contributory key distribution protocols that do not activate broadcasts. In order to show that this bound is sharp, we introduce a DH-based protocol that matches the bound in Section 3. With respect to the minimum number of total messages required by a key distribution protocol without broadcasting, we could reformulate the gossip problem by replacing the phone calls by letters or postcards. For the gossip problem in terms of messages, the following lemma can be proved.

Lemma 1 *Let $h(n)$ denote the minimum number of messages (letters) required by the gossip problem among n parties. Then it holds that $h(n) = 2 \cdot n - 2$.*

Proof It is not difficult to see that $h(n) \leq 2 \cdot n - 2$; for instance, all parties may send their information to one party, which subsequently distributes the entire information to everybody. Therefore it remains to be shown that $h(n) \geq 2 \cdot n - 2$, which can be done by induction on n.

Obviously $h(1) = 0$ and $h(2) = 2$. Assume there exists a protocol A to distribute the information of $n + 1$ parties $P_1, \ldots, P_{n+1}$ with the help of $2 \cdot n - 1$ messages. There has to be a party P_i whose first message M contains only the information originally held by P_i. We now show that protocol A can be modified to protocol A', which performs the entire information distribution between the parties $P_1, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n$ in fewer than $2 \cdot n - 2$ messages. This contradicts the induction assumption; hence $h(n+1) \geq 2 \cdot n = 2 \cdot (n+1) - 2$.

Now, if P_i does not send any further messages besides M, then A' can be derived from A by omitting M and all messages addressed to P_i. As P_i has to receive at least one message, protocol A' provides the entire gossip between $P_1, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n$ in fewer than $2 \cdot n - 2$ messages. Otherwise, if P_i sends a second message M' to party P_j, then A' can be constructed from A by omitting messages M and M' and letting P_j act for P_i. This again provides a gossip protocol for $P_1, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n$ requiring fewer than $2 \cdot n - 2$ messages. □

Finally, a lower bound for the number of simple rounds required by a contributory key distribution protocol without broadcasts can also be obtained by activating the information distribution argument. As every party is allowed to send and receive only one message per simple round, the number of parties who have a certain piece of information can at most be doubled in each round; i.e., at least d simple rounds are needed to let 2^d parties have a piece of information. Therefore, the number of simple rounds required to gossip information between n parties cannot be less than $\lceil log_2 n \rceil$, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x (for the gossip problem it can be shown that the lower bound for odd n is $\lceil log_2 n \rceil + 1$; see [Kn75]). From these results and observations we derive the following bounds for contributory key distribution systems without broadcasting.

Theorem 1 (without broadcasts) *Let $\mathcal{P}$ be a contributory group key distribution system for n parties not using broadcasts.*

1. *For the total number of messages $\varphi_1(\mathcal{P})$ required by $\mathcal{P}$ it holds that*

$$\varphi_1(\mathcal{P}) \geq 2 \cdot n - 2.$$

2. *For the total number of exchanges $\varphi_2(\mathcal{P})$ required by $\mathcal{P}$ and $n \geq 4$ it holds that*

$$\varphi_2(\mathcal{P}) \geq 2 \cdot n - 4.$$

3. *For the number of simple rounds $\varphi_3(\mathcal{P})$ required by $\mathcal{P}$ it holds that*

$$\varphi_3(\mathcal{P}) \geq \lceil log_2 n \rceil.$$

The situation changes if broadcasting is allowed. For example, Steiner et al. [STW96] introduced a DH-based contributory group key distribution protocol that requires only

$n-1$ messages and one broadcast. Obviously a total of n messages is optimal because at least every party has to disclose its piece of information. Lower bounds for the different measures of communication complexity for key distribution systems including broadcasts are summarized in the following theorem.

Theorem 2 (with broadcasts) *Let $\mathcal{P}$ be a contributory group key distribution system for n parties allowing broadcasts.*

1. *For the total number of messages $\psi_1(\mathcal{P})$ required by $\mathcal{P}$ it holds that*

$$\psi_1(\mathcal{P}) \geq n.$$

2. *For the total number of exchanges $\psi_2(\mathcal{P})$ required by $\mathcal{P}$ and $n \geq 3$ it holds that*

$$\psi_2(\mathcal{P}) \geq n.$$

3. *For the number of simple rounds $\psi_3(\mathcal{P})$ required by $\mathcal{P}$ it holds that*

$$\varphi_3(\mathcal{P}) \geq \lceil log_2 n \rceil.$$

4. *For the number of synchronous rounds $\psi_4(\mathcal{P})$ required by $\mathcal{P}$ it holds that*

$$\varphi_4(\mathcal{P}) \geq 1.$$

Proof Proposition 2 may be shown by proving that at least n exchanges or broadcasts are needed to distribute the n individual pieces of information (gossiping via broadcasts). This claim is obviously true for $n = 3$. Assume there exists an algorithm A to distribute the information of $n > 3$ parties which requires fewer than n exchanges and broadcasts. Then at least one exchange between two parties P_i and P_j has to be performed in order to enable every party to disclose its information. But now, by identifying the parties P_i and P_j, an algorithm A' can be derived from A, that performs the gossiping for $n-1$ parties with fewer than $n-1$ exchanges and broadcasts, which contradicts the induction assumption.

Proposition 3 holds because every party receives at most one message per simple round and therefore can hold at most 2^d messages after round d. □

3 Efficient Diffie–Hellman-Based Protocols

The aim of this section is to demonstrate that the lower bounds formulated for protocol complexity are realistic for Diffie–Hellman-based key distribution systems. Therefore, we summarize and introduce Diffie–Hellman-based key distribution protocols, which are efficient with respect to at least one of the above measures of protocol complexity.

3.1 Protocols requiring a minimum number of messages

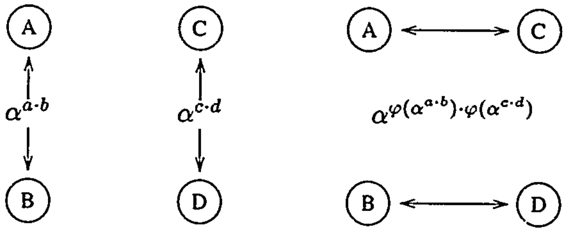
Examples for DH-based key distribution protocols requiring a minimum number of messages may be found in [STW96]. The protocol *GHD.1* by Steiner et al. is straightforward, without broadcasting, and performs with a total of $2 \cdot n - 2$ messages. The second protocol they introduce, denoted *GDH.2*, requires n messages to be sent, one of which is a broadcast.

protocol	GHD.1	GHD.2
messages	$2 \cdot n - 2$	n
exchanges	$2 \cdot n - 2$	n
simple rounds	$2 \cdot n - 2$	n
syn. rounds	$2 \cdot n - 2$	n
bc	–	1

3.2 Minimizing the number of exchanges: The octopus protocol

We now introduce a protocol without broadcasting that requires only $2 \cdot n - 4$ exchanges. For the broadcasting case, no further protocol has to be introduced because the lower bounds for the number of exchanges and the number of messages are both n. Therefore, the protocol GDH.2 also proves that the lower bound for the number of exchanges is sharp if broadcasting is possible.

In order to describe the subsequent Diffie–Hellman-based protocols, let G be a finite cyclic group of order q and let α be a generator of G (e.g., Diffie and Hellman [DH72] use $G = \mathbb{Z}_p^*$, where p is a prime). Furthermore, we assume that the individual participants choose their random secrets from $\mathbb{Z}_q$. A basic idea of the following protocol is to use a Diffie-Hellman key computed in one round as a random input for the subsequent round. Therefore, we further have to assume that there is a bijection $\varphi : G \longrightarrow \mathbb{Z}_q$, which has a short description. Whether there are appropriate bijections from G into $\mathbb{Z}_q$ depends on the group G. In the case that $G = \mathbb{Z}_p^*$, there is obviously no problem. But if G is supposed to be a subgroup of a much larger $\mathbb{Z}_{q'}$, this problem has to be further studied for particular groups G and practical solutions have to be found.


A
C
A
C
$\alpha^{a\cdot b}$
$\alpha^{c\cdot d}$
$\alpha^{\varphi(\alpha^{a\cdot b})\cdot\varphi(\alpha^{c\cdot d})}$
B
D
B
D

Before introducing the octopus protocol, we first observe that four parties A, B, C, and D may generate a group key using only four exchanges. First, parties A and B and parties C and D perform a Diffie–Hellman key exchange generating keys $\alpha^{a\cdot b}$ and $\alpha^{c\cdot d}$, respectively. Subsequently, A and C as well as B and D carry out a Diffie–Hellman key exchange using as secret values the keys generated in the first step; i.e., A (B) sends $\alpha^{\varphi(\alpha^{a\cdot b})}$ to C (to D) while C (D) sends $\alpha^{\varphi(\alpha^{c\cdot d})}$ to A (to B) such that A and C (B and D) can generate the joint key $\alpha^{\varphi(\alpha^{a\cdot b})\cdot\varphi(\alpha^{c\cdot d})}$.

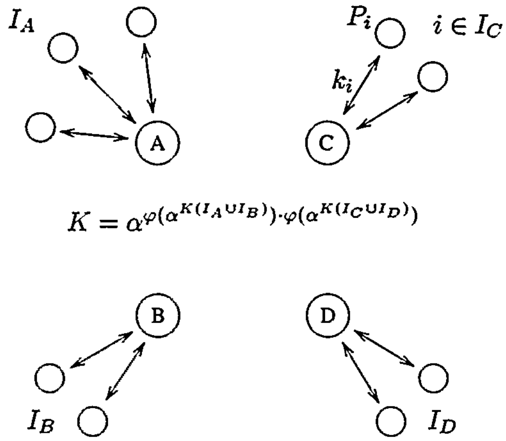
In the **octopus protocol**, participants $P_1, \ldots, P_n$ generate a common group key by first dividing themselves into five groups. Four participants $P_{n-3}, P_{n-2}, P_{n-1}, P_n$ take charge of the central control; we denote these participants A, B, C, and D, respectively. The remaining parties distribute themselves into four groups $\{P_i \mid i \in I_A\}$, $\{P_i \mid i \in I_B\}$, $\{P_i \mid i \in I_C\}$, and $\{P_i \mid i \in I_D\}$, where I_A, I_B, I_C, and I_D are pairwise disjoint, possibly of equal size, and $I_A \cup I_B \cup I_C \cup I_D = \{1, \ldots, n-4\}$. Now $P_1, \ldots, P_n$ can generate a group key as follows:

1. For all $X \in \{A, B, C, D\}$ and all $i \in I_X$, the party X generates a joint key k_i with P_i by performing a Diffie–Hellman key exchange.

2. The participants A, B, C, and D perform the 4-party key exchange described above using the values $a = K(I_A)$, $b = K(I_B)$, $c = K(I_C)$, and $d = K(I_D)$, where $K(J) := \prod_{i \in J} \varphi(k_i)$ for $J \subseteq \{1, \dots, n-4\}$. Thereafter, A, B, C, and D hold the joint and later group key
$$K := \alpha^{\varphi(\alpha^{K(I_A \cup I_B)}) \cdot \varphi(\alpha^{K(I_C \cup I_D)})}.$$

3. We describe this step only for A; the parties B, C, and D act correspondingly. For all $j \in I_A$, the participant A sends the following two values to P_j:
$$\alpha^{K(I_B \cup I_A \setminus \{j\})} \quad \text{and} \quad \alpha^{\varphi(\alpha^{K(I_C \cup I_D)})}.$$
Now P_j is able to generate K; first P_j calculates $(\alpha^{K(I_B \cup I_A \setminus \{j\})})^{\varphi(k_j)} = \alpha^{K(I_A \cup I_B)}$ and then $K = (\alpha^{\varphi(\alpha^{K(I_C \cup I_D)})})^{\varphi(\alpha^{K(I_A \cup I_B)})}$.

This protocol requires $n - 4$ exchanges to generate the DH keys k_i, four exchanges for the key agreement between A, B, C, and D, and finally $n - 4$ messages to be sent from A, B, C, D to $P_1, \dots, P_{n-4}$. Hence the protocol performs a minimum number of $2 \cdot n - 4$ exchanges.



I_A
A
P_i
$i \in I_C$
k_i
C
$K = \alpha^{\varphi(\alpha^{K(I_A \cup I_B)}) \cdot \varphi(\alpha^{K(I_C \cup I_D)})}$
B
D
I_B
I_D

protocol	octopus
messages	$3 \cdot n - 4$
exchanges	$2 \cdot n - 4$
simple rounds	$2 \cdot \lceil \frac{n-4}{4} \rceil + 2$
syn. rounds	4
bc	–

3.3 Minimizing the number of rounds: The 2^d-octopus protocol

The number of simple rounds can be minimized by generalizing the idea of the 4-party key agreement described above. In general, 2^d parties can agree upon a key within d simple rounds by performing DH key exchanges on the edges of a d-dimensional cube.

In order to describe formally the **cube protocol for 2^d participants**, we identify the 2^d participants with the vectors of the d-dimensional vector space $GF(2)^d$ and choose a basis $\vec{b}_1, \dots, \vec{b}_d$ of $GF(2)^d$. Now the protocol may be performed in d rounds as follows:

1. In the first round, every participant $\vec{v} \in GF(2)^d$ generates a random number $r_{\vec{v}}$ and performs a DH key exchange with participant $\vec{v} + \vec{b}_1$ using the values $r_{\vec{v}}$ and $r_{\vec{v}+\vec{b}_1}$, respectively.

i. In the i-th round, every participant $\vec{v} \in GF(2)^d$ performs a DH key exchange with participant $\vec{v}+\vec{b}_i$, where both parties use the value generated in round $i-1$ as the secret value for the key exchange.

In every round, the participants communicate on a maximum number of parallel edges of the d-dimensional cube (in round i in the direction $\vec{b}_i$); thus, every party is involved in exactly one DH exchange per round. Furthermore, all parties share a common key at the end of this protocol because the vectors $\vec{b}_1, \dots, \vec{b}_d$ form a basis of the vector space $GF(2)^d$. This cube pattern is also used in [Kn75] to manage the gossip problem with a minimum number of rounds. [Bu90] suggests using parallel classes in more general geometric structures to distribute information between n parties, which might as well serve as a basis for group key distribution protocols.

In order to formulate a protocol for an arbitrary number of participants ($\neq 2^d$), which requires a low number of simple rounds, the idea of the octopus protocol can be adopted again. In the **2^d-octopus protocol** the participants act as in the octopus protocol introduced above with the only difference that 2^d instead of four parties are distinguished to take charge of the central control, whereas the remaining $n - 2^d$ parties divide into 2^d groups. In other words, in steps 1 and 3 of the octopus protocol, 2^d participants manage communication with the rest and in step 2 these 2^d parties perform the cube protocol for 2^d participants. If the number of participants is n and if d is the largest integer smaller than $log_2 n$, then the 2^d-octopus protocol requires $1 + d + 1 = \lceil log_2 n \rceil + 1$ simple rounds. In general, we obtain the following values of protocol complexity for the cube and the 2^d-octopus protocol.

protocol	2^d-cube	2^d-octopus
messages	$n \cdot d$	$3 \cdot (n - 2^d) + 2^d \cdot d$
exchanges	$n \cdot d/2$	$2 \cdot (n - 2^d) + 2^{d-1} \cdot d$
simple rounds	d	$2 \cdot \lceil \frac{n-2^d}{2^d} \rceil + d$
syn. rounds	d	$2 + d$
bc	–	–

The 2^d-octopus protocol is especially interesting because it provides a tradeoff option between the total number of messages or exchanges needed and the number of rounds. For $d = 2$ (octopus protocol) the number of exchanges is optimal, whereas the number of simple rounds is comparatively high. On the other hand, if d satisfies $2^{d-1} < n \leq 2^d$, the number of simple rounds required is very low and the total number of messages is high. Furthermore, the protocol enables the group to decide how many participants should share control of the protocol.

With the 2^d-octopus protocols we have introduced a class of key distribution systems without broadcasting which matches the lower bound $\lceil log_2 n \rceil$ for the total number of simple rounds if n is a power of 2; otherwise the protocols require $\lceil log_2 n \rceil + 1$ simple rounds. Furthermore, the 1-octopus protocol ($d = 0$) is a protocol that requires two synchronous rounds, which is at least close to optimal. In other words, with respect to the bounds formulated for the number of rounds required by contributory key distribution protocols, we have formulated protocols that exceed the bounds by at most one round. It remains an open question whether there exist, for $n \neq 2^d$, protocols (with or without broadcasting) that require only $\lceil log_2 n \rceil$ simple rounds. Another interesting question is whether one can formulate a contributory key distribution system that requires only one synchronous round.

The protocols introduced should, of course, be elaborated further, for example by including authentication procedures. But here our main objective was to clarify systematically the question of protocol complexity rather than to design detailed protocols. Therefore we do not discuss refinements of those protocols further and conclude this paper with a security analysis of the protocols.

4 Security of the Protocols

It was claimed in the previous section that the protocols introduced are Diffie–Hellman-based, which remains to be proved. The main new building block of those protocols is the cube protocol for 2^d participants. Therefore, we restrict our discussion to proving that the cube protocol for 2^d participants is Diffie–Hellman-based; a very similar proof applies to the 2^d-octopus protocol.

In the following, it has to be shown that the key generated by the cube protocol cannot be distinguished by a polynomial algorithm from a random number if all values transmitted during a protocol run are known. This claim has to be derived under the assumption that the same holds for the 2-party DH protocol. The method of the subsequent proof is similar to that used in [STW96] to show that the class of generic DH protocols is Diffie–Hellman-based.

Let G denote again the group underlying the protocol; let q be the order of G, α a generator of G, and φ a bijection from G into $\mathbb{Z}_q$. Furthermore let d be a positive integer and $X = (N_1, \dots, N_{2^d})$ randomly chosen from $\mathbb{Z}_q^{2^d}$. Now we consider a cube protocol with 2^d participants $P_1, \dots, P_{2^d}$, where P_i has chosen N_i as the secret starting value. Then

- $\nu(d, X)$ denotes the ordered 2^d-tuple of all values transmitted during the performance of the cube protocol (we presume a fixed order) and
- $K(d, X)$ denotes the final common key generated by the cube protocol with starting values $X = (N_1, \dots, N_{2^d})$.

Now we consider the probability distribution $A_d := (\nu(d, X), y)$ obtained by the probability of $(\nu(d, X), y)$ if $X \in \mathbb{Z}_q^{2^d}$ and $y \in G$ are randomly chosen. Furthermore we define the probability distribution $F_d := (\nu(d, X), K(d, X))$ obtained if we chose X randomly from $\mathbb{Z}_q^{2^d}$. The polynomial indistinguishability between two probability distributions is denoted by $\approx_{poly}$. Then the claim that the cube protocol is Diffie–Hellman-based can be expressed as in the following theorem.

Theorem 3 *$A_1 \approx_{poly} F_1$ implies $A_d \approx_{poly} F_d$ for every positive integer d.*

Proof [sketch] The implication claimed in the theorem may be proved by induction on d. First we observe that one can rewrite $\nu(d, X)$ with $X = (N_1, \dots, N_{2^d})$ as a permutation of

$$(\nu(d-1, X_1), \nu(d-1, X_2), \alpha^{K(d-1,X_1)}, \alpha^{K(d-1,X_2)}),$$

where $X_1 = (N_1, \dots, N_{2^{d-1}})$ and $X_2 = (N_{2^{d-1}+1}, \dots, N_{2^d})$. Furthermore, it holds that

$$K(d, X) = \varphi(\alpha^{K(d-1,X_1)\cdot K(d-1,X_2)}).$$

In order to show $A_d \approx_{poly} F_d$, we define probability distributions B_d, C_d, D_d, and E_d on $\mathbb{Z}_q^{2^d} \times G$ with $A_d \approx_{poly} B_d \approx_{poly} C_d \approx_{poly} D_d \approx_{poly} E_d \approx_{poly} F_d$, which implies $A_d \approx_{poly} F_d$. All of the following probability distributions are defined on $\mathbb{Z}_q^{2^d} \times G$ by randomly choosing X_1, X_2 from $\mathbb{Z}_q^{2^{d-1}}$ and $y, c_1, c_2 \in G$:

$$\begin{aligned}
A_d &= (\nu(d-1, X_1), \nu(d-1, X_2), \alpha^{K(d-1,X_1)}, \\
&\qquad \alpha^{K(d-1,X_2)}, y), \\
B_d &= (\nu(d-1, X_1), \nu(d-1, X_2), \alpha^{c_1}, \alpha^{K(d-1,X_2)}, y), \\
C_d &= (\nu(d-1, X_1), \nu(d-1, X_2), \alpha^{c_1}, \alpha^{c_2}, y), \\
D_d &= (\nu(d-1, X_1), \nu(d-1, X_2), \alpha^{c_1}, \alpha^{c_2}, \alpha^{c_1 \cdot c_2}), \\
E_d &= (\nu(d-1, X_1), \nu(d-1, X_2), \alpha^{c_1}, \alpha^{K(d-1,X_2)}, \\
&\qquad \alpha^{c_1 \cdot K(d-1,X_2)}), \\
F_d &= (\nu(d-1, X_1), \nu(d-1, X_2), \alpha^{K(d-1,X_1)}, \\
&\qquad \alpha^{K(d-1,X_2)}, \alpha^{K(d-1,X_1)\cdot K(d-1,X_2)}).
\end{aligned}$$

Proposition 1: $A_{d-1} \approx_{poly} F_{d-1}$ implies $A_d \approx_{poly} B_d$.

Assume that there exists a polynomial algorithm that can distinguish between A_d and B_d. We show that this algorithm can be used to distinguish between A_{d-1} and F_{d-1} in polynomial time as well. Let $(\nu(d-1, Z), z)$ be an instance of $A_{d-1} \approx_{poly} F_{d-1}$. Then we consider the instance $\mathcal{X} = (\nu(d-1, Z), \nu(d-1, X_2), \alpha^z, \alpha^{K(d-1,X_2)}, y)$ of $A_d \approx_{poly} B_d$. We observe that $(\nu(d-1, Z), z)$ belongs to F_{d-1} iff $\mathcal{X}$ belongs to A_d. On the other hand, if $(\nu(d-1, Z), z)$ belongs to A_{d-1}, then $\mathcal{X}$ belongs to B_d and conversely. This provides an polynomial algorithm to distinguish between A_d and B_d.

Proposition 2: $A_{d-1} \approx_{poly} F_{d-1}$ implies $B_d \approx_{poly} C_d$ and $D_d \approx_{poly} E_d \approx_{poly} F_d$.

This can by proved analogously to Proposition 1.

Proposition 3: $A_1 \approx_{poly} F_1$ implies $C_d \approx_{poly} D_d$.

Again we assume that there exists a polynomial algorithm that can distinguish between C_d and D_d. Now let (u, v, w) be an instance of $A_1 \approx_{poly} F_1$. Then we may construct an instance of $C_d \approx_{poly} D_d$ by $\mathcal{X} = (\nu(d-1, X_1), \nu(d-1, X_2), u, v, w)$ for which the following holds: if (u, v, w) belongs to A_1, then $\mathcal{X}$ belongs to C_d, and if (u, v, w) belongs to F_1, then $\mathcal{X}$ belongs to D_d. This completes the proof. □

Acknowledgements: We wish to thank Victor Shoup, Michael Steiner, Gene Tsudik, and Michael Waidner for valuable comments and fruitful discussions.

References

[ABM87] Alon, N., Barak, A., Manber, U., On disseminating information reliably without broadcasting. In *IEEE Proceedings of the 7^{th} International Conference on Distributed Computing Systems.* Berlin, September 1987, 74–81.

[BM93] Babaoğlu, Ö., Marzullo, K., Consistent Global State of Distributed Systems: Fundamental Concepts and Mechanisms. In S. Mullender (ed.), *Distributed Systems.* NY: ACM Press, 1993, 55–145.

[BS72] Baker, B., Shostak, R., Gossips and Telephones. *Discrete Mathematics*, 4, 1972, 191–193.

[Be97] Becker, K., *Design und Analyse von Konferenzschlüsselsystemen.* PhD Dissertation, Giessen 1996.

[Be72] Berman, G., The Gossip Problem. *Discrete Mathematics*, 4, 1972, 91.

[Bu90] Beutelspacher, A., How to Communicate Efficiently. *Journal of Combinatorial Theory, Series A*, **54**, 1990, 312–316.

[BD94] Burmester, M., Desmedt, Y., A secure and efficient conference key distribution system. In A. De Santis (ed.), *Advances in Cryptology–EUROCRYPT '94*, LNCS 950, Berlin: Springer 1994, 275–286.

[DH72] Diffie, W., Hellman, M., New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 1976, 644–654.

[ITW82] Ingemarson, I., Tang, D., Wong, C., A conference key distribution system. *IEEE Transactions on Information Theory*, 28(5), 1982, 714–720.

[Kn75] Knödel, W., New Gossips and Telephones. *Discrete Mathematics*, **13**, 1975, 95.

[STW96] Steiner, M., Tsudik, G., Waidner, M., Diffie–Hellman Key Distribution Extended to Groups. 3rd ACM Conference on Computer and Communications Security, ACM Press, 1996, 31-37.