Anonymous Oblivious Transfer

J. Müller-Quade and H. Imai
Imai Laboratory, Institute of Industrial Science, The University of Tokyo
(December 3^{rd}, 2000)

In this short note we want to introduce *anonymous oblivious transfer* a new cryptographic primitive which can be proven to be strictly more powerful than oblivious transfer. We show that all functions can be robustly realized by multi party protocols with *anonymous oblivious transfer*. No assumption about possible collusions of cheaters or disruptors have to be made.

Furthermore we shortly discuss how to realize *anonymous oblivious transfer* with oblivious broadcast or by quantum cryptography. The protocol of *anonymous oblivious transfer* was inspired by a quantum protocol: the *anonymous quantum channel*.

I. INTRODUCTION

In [2,8,6] multi party protocols with oblivious transfer were presented which can tolerate a dishonest majority. These protocols work with perfect security if all players cooperate. But already one disruptor can abort the protocol without being detected. The contribution of [1] were protocols more robust against disruption. The idea was to replace two party subprotocols which failed by multi party protocols. Then either these protocols did work or a cheater could be identified.

Unfortunately replacing an oblivious transfer where the sender or the receiver refuses to coopertate by a multi party protocol weakens the security of the protocol. In [1] we can observe a trade off between the size of a tolerable collusion of active cheaters (including disruptors) and the size of a collusion of passive cheaters unable to obtain secret data.

In this paper we present the new cryptographic primitive *anonymous oblivious transfer* and prove that it is strictly more powerful than oblivious transfer. With this primitive we can realize multi party protocols which work with perfect security or a cheater can be identified unambigiously. As we cannot expect higher robustness and security than that we claim that anonymous oblivious transfer is the most powerful cryptographic primitive which can achieve unconditional security. We recently learned about independent work in this direction carried out by [7].

II. MULTI PARTY PROTOCOLS

In a multi party protocol a set P of players wants to correctly compute a function $f(a_1, \ldots, a_n)$ which depends on secret inputs of n players. Some players might collude to cheat in the protocol as to obtain information about secret inputs of the other players or to modify the result of the computation. Possible collusions of cheaters are modelled by *adversary structures*

Definition 1 *An adversary structure is a monotone set $\mathcal{A} \subseteq 2^P$, i. e., for subsets $S' \subseteq S$ of P the property $S \in \mathcal{A}$ implies $S' \in \mathcal{A}$.*

We assume that one set $A \in \mathcal{A}$ of players collude to cheat in the protocol. These players take all their action based on their common knowledge.

The main properties of a multi party protocol are:

1. A multi party protocol is said to be $\mathcal{A}$-*secure* if no single collusion from $\mathcal{A}$ is able to obtain information about the secret inputs of other participants which cannot be derived from the result and the inputs of the colluding players.
2. A multi party protocol is $\mathcal{A}$-*partially correct* if no possible collusion can let the protocol terminate with a wrong result.
3. A multi party protocol is called $\mathcal{A}$-*fair* if no collusion from $\mathcal{A}$ can reconstruct the result of the multi party computation earlier then all honest participants together. No collusion should be able to run off with the result.

We will be more strict here and demand robustness even against disruptors.

2' A multi party protocol is $\mathcal{A}$-*correct* whenever no single collusion from $\mathcal{A}$ can abort the protocol, modify its result, or take actions such that some player gets to know a secret value.

A protocol is called $\mathcal{A}$-*robust* if it has all of the above properties. Note that we will allow only one collusion to cheat, but we think of every single player as being curious, i. e., even if he is not in the collusion actually cheating he will eavesdrop all information he can obtain without being detected cheating

With oblivious transfer all multi party protocols can be realized with perfect security if all players are cooperating [2,8,6]. But a collusion of players can abort the calculation, see next section.

III. IMPOSSIBILITY RESULTS

In this section we show that oblivious transfer is not able to implement all multi party protocols in the presence of cheaters which can derivate arbitrarily from the protocol. Not even together with a broadcast channel. Protocols offering perfect secrecy of the inputs can be aborted by a collusion of players.

Lemma 2 *Let P be a set of players for which each pair of players is connected by a secure and authenticated oblivious transfer channel and each player has access to a*

broadcast channel. Then $\mathcal{A}$-robust multi party computations are possible for all functions if and only if no two sets of $\mathcal{A}$ cover $P \setminus \{P_i\}$ for a player $P_i \in P$ or $|P| = 2$.

Proof: Let A and B be two possible collusions covering $P \setminus \{P_i\}$, then oblivious transfer cannot be implemented $\mathcal{A}$-robustly between players of A and players of B. Between any two players Alice $\in A$ and Bob $\in B$ the oblivious transfer channel does not work, but it is not obvious for the player P_i who is refusing to cooperate. The player P_i must assist Alice and Bob. As no other player can assist we are in the three party situation with an oblivious transfer channel only between Alice and P_i and Bob and P_i. For each bit being transferred from Alice to Bob the player P_i knows either as much as Alice about this bit or he knows as much as Bob. The players Alice and Bob cannot agree on a bit known to both without P_i knowing it, too. Hence oblivious transfer from Alice to Bob becomes impossible without P_i having to learn a secret of Alice or a secret of Bob. □

IV. MULTI PARTY PROTOCOLS

In the multi party protocols of [6,1] a collusion of disruptors can abort the protocol if an assumption about possible collusions of disruptors is violated. We would like to have cryptographic primitives where every time a conflict arises a cheater can be identified. Two such primitives are *global bit commitment* and *undeniable oblivious transfer*. We will show in the following that these primitives, defined below, can realize the subprotocols needed in [6,1] relative to no assumptions about possible collusions.

Definition 3 *A* global bit commitment (GBC) *binds a player to all other players to the same bit in a way that this bit cannot be changed with a non negligible probability unless the player colludes with all other players.*

Definition 4 *An* undeniable oblivious transfer (UOT) protocol *from a player Alice $\in P$ to a player Bob $\in P$ allows Alice to generate a GBC for a bit b in a way that Bob learns the bit b with probability $1/2$ and Alice cannot know if Bob learned b.*

Now we introduce the notions used for the multi party protocols.

Definition 5 *A* global bit commitment with Xor *(*GBCX*) to a bit b is a GBC to bits b_{1L}, $b_{2L}, \ldots, b_{mL}$, $b_{1R}, \ldots, b_{mR}$ such that for each i $b_{iL} \oplus b_{iR} = b$.*

One important ability of these bit cimmitments with Xor is given in the next result, which is taken from [6], but see also references therein.

Theorem 6 *GBCX allow zero knowledge proofs of linear relations among several bits a player is committed to using GBCX. Especially (in)equality of bits or a bit string being contained in a linear code.*

Furthermore GBCXs can be copied, as proofs may destroy a GBCX.

Proof: We will not state a full proof here as it can be found in [6]. But we will restate the copying procedure as it is an important subprotocol of all of the following protocols.

Suppose Alice is committed to Bob to a bit b and wants two instances of this commitment. Then Alice ceates $3m$ pairs of global bit commitments such that each pair Xors to b. Then all other player, by coin tossing, randomly partition these $3m$ pairs in three subsets of m pairs, thus obtaining three GBCX and ask Alice to prove the equality of the first new BCX with her GBCX for b. This destroys the old GBCX and one of the new GBCX, but an honest Alice can thereby convince all players that the two remaining GBCX both stand for the value b. □

The basic building block for multi party protocols of [6] are distributed bit commitments, where each player is committed to a share of a bit.

Definition 7 *A* distributed bit commitment (DBC) of a user *Alice $\in P$ to a bit b consists of n GBCX one created by each player of P such that only Alice knows how to open all of them and the Xor of all values ot the GBCX equals b.*

An intermediate result DBC *consists of n GBCX such that no subset of players unequal P can know how to open all of the GBCX.*

Lemma 8 *With a protocol for generating GBCX and a broadcast channel one can realize a DBC of a user.*

Proof: Each player generates a GBCX and opens the commitment to Alice. In case of a conflict the player opens his GBCX publicly. Then Alice creates a GBCX such that the parity bit is the bit she wanted to create a DBC for. Only Alice knows how to open all commitments as she created one herself. □

The intermediate result DBCs are automatically generated by the multi party protocols for these we need the key protocol of [6].

Definition 9 *Given two players Alice and Bob where Alice is committed to bits b_0, b_1 and Bob is committed to a bit a. Then a* committed oblivious transfer *protocol (*COT*) is a protocol where Alice inputs her knowledge about her two commitments and Bob will input his knowledge about his commitment and the result will be that Bob is committed to b_a.*

In a global committed oblivious transfer *protocol (*GCOT*) all players are convinced of the validity of the commitments, i.e., that indeed Bob is committed to b_a after the protocol.*

For the next result we use one-out-of-two UOT, which is the usual one-out-of-two OT, but the sender is (by GBCs) committed to the two bits the receiver can choose from. The standard reduction from one-out-of-two OT to OT can be used to turn UOT into one-out-of-two UOT.

Lemma 10 *With UOT and an authenticated broadcast channel one can realize GCOT.*

Proof: We will essentially restate the GCOT protocol of [6] and see that with one-out-of-two UOT instead of one-out-of-two OT any conflict results in the identification of a cheater.

GCOT$(a_0, a_1)(b)$

1. All participants together choose one decodable $[m, k, d]$ linear code $\mathcal{C}$ with $k > (1/2 + 2\sigma)m$ and $d > \epsilon n$ for positive constants σ, ϵ, efficiently decoding t errors.
2. Alice randomly picks $c_0, c_1 \in \mathcal{C}$, committs to the bits c_0^i and c_1^i ($i \in \{1, \ldots, m\}$) of the code words, and proves that the codewords fulfil the linear relations of $\mathcal{C}$.
3. Bob randomly picks $I_0, I_1 \subset \{1, \ldots, M\}$, with $|I_0| = |I_1| = \sigma m$, $I_1 \cap I_0 = \emptyset$ and sets $b^i \leftarrow \overline{b}$ for $i \in I_0$ and $b^i \leftarrow b$ for $i \notin I_0$.
4. Alice runs UOT$(c_0^i, c_1^i)(b^i)$ with Bob who gets w^i for $i \in \{1, \ldots, m\}$. Bob tells $I = I_0 \cup I_1$ to Alice who opens c_0^i, c_1^i for each $i \in I$.
5. Bob checks that $w^i = c_{\overline{b}}^i$ for $i \in I_0$ and $w^i = c_b^i$ for $i \in I_1$, sets $w^i \leftarrow c_b^i$, for $i \in I_0$ and corrects w using $\mathcal{C}$'s decoding algorithm, commits to w^i for $i \in \{1, \ldots, m\}$, and proves that $w^1 \ldots w^m \in \mathcal{C}$.
6. All players together randomly pick a subset $I_2 \subset \{1, \ldots, m\}$ with $|I_2| = \sigma m$, $I_2 \cap I = \emptyset$ and Alice opens c_0^i and c_1^i for $i \in I_2$.
7. Bob proves that $w^i = c_b^i$ for $i \in I_2$.
8. Alice randomly picks and announces a privacy amplification function $h : \{0,1\}^m \to \{0,1\}$ such that $a_0 = h(c_0)$ and $a_1 = h(c_1)$ and proves $a_0 = h(c_0^1, \ldots, c_0^m)$ and $a_1 = h(c_1^1, \ldots, c_1^m)$.
9. Bob sets $a \leftarrow h(w)$, commits to a and proves $a = h(w^1 \ldots, w^m)$.

A conflict between Alice and Bob can only appear in connection with step 4 or step 5. If these two steps would be performed honestly then all other steps can be checked by all other players and it becomes immediately clear who is cheating. In a conflict in connection with step 4 or step 5 Bob claims that Alice sent something inconsistent over the oblivious transfer channel or Alice accuses Bob to not have committed to what he received.

In case of a conflict Alice opens all bits of c_0, c_1 to which she is committed by the UOT also she opens her GBCX to these codewords, if she is not able to do it or unveils non code words or other inconsistent information she is detected cheating. The bits of c_0, c_1 do not give away any secret as these are random code words. If Alices information is correctly unveiled and is consistent with all her past actions (proofs) then Bob was cheating if he did complain. If it was Alice complaining Bob has to prove zero knowledgly that the bit string w he is committed to equals c_0 or equals c_1 if he is able to convince all other players Alice is detected cheating (conflicts appearing during the proofs can be resolved easily as it is obvious for every player who is cheating). □

One other important property of multi party protocols is *fairness*. A multi party protocol is called *fair* if no collusion of players can reconstruct the result of the protocol earlier than all honest players. This problem is solved in the literature [4,8] and will not be discussed here.

Hence we have everything to follow the protocols of [6] robustly and in the following we need only to prove that a certain cryptographic primitive can realize GBC (or GBCX) and UOT and we know that it is capable of realizing all multi party protocols with perfect security and robustness.

Theorem 11 *Given a set of players P such that every player can generate global bit commitments and we have an undeniable oblivious transfer between every pair of players. Then all functions can be computed 2^P-robustly by multi party protocols.*

Proof: First we note that we do not need a broadcast channel as generating a GBC and unveiling it can be viewed as broadcasting. We now sketch the phases of a multi party protocol following [6]. To implement oblivious circuit evaluation to realize arbitrary functions we have to show the existence of an AND and a NOT function on DBCs and clearify how a protocol is initialized and how it is ended.

Initialization Phase: All players have to agree on the function to be computed as well as on the circuit F to be used, they have to agree on an adversary structure $\mathcal{A}$ such that the protocol will be $\mathcal{A}$ robust and all players have to agree on the security parameters used and on a code $\mathcal{C}$ for the GCOT protocol.

Then all players create DBCs to commit to their inputs.

Computing Phase: The circuit is evaluated using AND and NOT gates on the input DBCs.

An AND on commitments can be realized by the following protocol: Alice is committed to a and Bob is committed to b. Then Alice chooses a random bit a' and runs GCOT$(a', a' \oplus a)(b)$ with Bob who gets b'. We have $a' \oplus b' = a \wedge b$ because for $b = 0$ we have $b' = a'$ and hence $a' \oplus b' = 0$, for $b = 1$ we get $b' = a \oplus a'$ and $a' \oplus b' = a$.

To evaluate an AND on DBCs we observe that $(\bigoplus_{i=1}^n a_i) \wedge (\bigoplus_{j=1}^n b_j) = \bigoplus_{i,j=1}^n (a_i \wedge b_j)$. From this we can conclude that an AND operation on DBCs can be realized by n^2 GPAND one for each pair of players and Xor operations for each player.

To implement the NOT gate one player is picked who must invert his "share". This players generates a new GBCX and proves that it is unequal to the GBCX he held before. Note that the GCOT within the AND protocol has to work only in one direction between every pair of

players. Sometimes one needs several copies of a DBC. A DBC is copied by copying the GBCX it consists of. A GBCX can be copied by copying all its BCX with the procedure of Theorem 6.

Revelation Phase: The result of a computation is hidden in DBCs. These have to be unveiled in a way to ensure the fairness of the protocol. Following [6] we use the techniques from [4,8] to gradually unveil the secret information such that no collusion can run off with an advantage of more than a fraction of a bit. Of course an $\widetilde{\mathcal{A}}$-secure protocol cannot be more than $\widetilde{\mathcal{A}}$-fair. □

V. ANONYMOUS OBLIVIOUS TRANSFER

We next define *anonymous oblivious transfer*.

Definition 12 *An* anonymous oblivious transfer *(*AOT*) protocol allows a player Alice $\in P$ to send a bit string $b_1 \dots b_m$ to a player Bob $\in P$ such that Bob receives each bit of the bit string with probability* $1/2$ *or he receives* $\perp$ *which indicates that he will not learn this bit. Alice cannot know which bits Bob received. Furthermore Bob does not know which player sent the bit string.*

For the following we will need some subprotocols which can easily be realized by AOT. To realize them we need a message authentication function $\mathrm{Auth}(x, y)$ which outputs a string which authenticats the message x with the secret y, see [11,3] for an unconditional signature scheme based on such a function and anonymous transfer.

Lemma 13 *With AOT one can realize an authenticated broadcast channel.*

Proof: Every player sends l times anonymously a random number to Alice. Alice sends her message m to every player together with $\mathrm{Auth}(m, r)$ for all random numbers r Alice received. [1] Then every pair of players compares the message they received. Either they are all the same and the protocol was successful or two different messages show up (one might be the empty message). Now two cases can happen:

1. The second message is correctly authenticated, then we have a high probability (depending on l) that the sender Alice was cheating or
2. the second message is not correctly authenticated.

In both cases we repeat the protocol until one of the following cases holds:

1. The protocol was successful.
2. Alice is in conflict with all other players and has to leave the protocol.
3. The players complaining about Alice are always the same, then these must be cheating as Alice cannot know who sent which random number.
4. Enough different correctly authenticated messages are found such that the probability that Alice is cheating is above a certain threshold and she is expelled from the protocol.

□

[1] This can be seen as "signing" the message [11,3].

Lemma 14 *With AOT one can realize anonymous message transfer and an anonymous broadcast channel which can fail only n times or someone leaves the protocol.*

Proof: To send a message anonymously one has to encode the message with an error correcting code to cope with the erasures of the AOT.

For an anonymous broadcast Alice sends her message m anonymously to a player P_i. This player broadcasts the message. If he broadcasts something wrong Alice is in conflict with this player, complains about him using the authenticated broadcast, and picks another player P_j to start the procedure anew. Either the anonymous broadcast will eventually be successfull or Alice will leave the protocol as she is in conflict with all other players. □

Corollary 15 *With AOT one can realize the anonymous message transfer and anonymous broadcast of Lemma 14 in a way that the anonymous sender can later identify himself.*

Proof: For an anonymous broadcast with later identification Alice authenticates her message m with n random numbers which she sends anonymously to the players. Each player receives one random number.

Then she anonymously broadcasts the thus authenticated message according to Lemma 14. No other player is later able to impersonate Alice as only she knows the secret random numbers of the honest players. □

With these protocols we can realize GBCX.

Lemma 16 *With AOT one can realize GBCX.*

Proof: We let all players create GBCX according to the protocol of [6], but anonymously, using AOT and anonymous broadcast. Then after some time no new conflicts occur for l anonymous GBCX of each player (l is a security parameter which is polynomial in n). If a player Alice was unable to create a GBCX we will split the set of players in a way that one set contains all honest players and the other sets contain only cheaters. We explain this in more detail by the two cases which can occur:

1. If Alice was honest then, as a cheater cannot distinguish between the honest players after some time if the cheater keeps complaining about Alice this

cheater will be in conflict with all honest players. Furthermore all honest players will know it. Now we can seperate the set P of players several subsets such that all players in each subset are in conflict with the same players. Then we can be sure that one of the sets contains all honest players and every honest player knows it.

2. If Alice was dishonest then we will also seperate the set P. Alice will be in one group with all players complaining about the same players as Alice did (these are all honest players if Alice were honest) all other players will be in the other sets. As Alice is a cheater and hence in conflict with an honest player all players in her group must be cheaters, too.

□

Note that the protocol to create GBCX for all players needs only polynomial time in n, as only n^2 conflicts are possible.

After having realized GBCX we need to implement UOT.

Lemma 17 *With AOT one can realize UOT.*

Proof: Alice creates a GBCX following Lemma 16 and Bob publishes positions of two substrings of the strings Alice sent to him. One substring where he knows all the bits and one substring where he knows nothing. The substrings must have approximately the same length.

Alice publishes the bits of one of the substrings. Then Bob either learnt nothing new or he knows the bit Alice is committed to. We have realized UOT if we can show that no other player learns the bit Alice is committed to by the information published by Alice, but this is trivial as Alice sent different strings to different players. □

VI. REALIZING AOT

A. Quantum Protocols

Anonymous oblivious transfer was inspired by a quantum protocol [9]. But it cannot be realized by a quantum protocol unless no two possible collusions cover the set P of players.

The idea for the realization is to follow normal quantum multi party protocols [10] if not two sets covering $P \setminus P_i$ are in conflict. In case of such a conflict the player P_i is not a disruptor or active cheater by assumption. This player can now forward quantum information between the two sets which are in conflict. Quantum cryptography allows to keep the player P_i from eavesdropping the quantum data excluding what happened in Lemma 2. As the player P_i can forward all quantum information in the same way and send quantum information himself this realizes an *anonymous quantum channel*. Together with the results of [10] we get:

Theorem 18 *Robust quantum multi party protocols for all functions are possible if and only if no two possible collusions cover the set P of players.*

These protocols become robust against a set of possible collusions after termination which may contain one and only one complement of a collusion tolerable during the execution of the protocol.

For a proof see [9].

Especially a quantum channel can be more powerfel than oblivious transfer (See Lemma 2). For details please refer to [10,9].

B. Oblivios Broadcast

We can think of each player broadcasting weak signals. Signals which can be received only with a certain probability which is independent for all receiving players. In this subsection we want to show that this primitive is equally powerful as AOT.

Definition 19 *An* oblivious broadcast *channel is a protocol where a player inputs a bit string and every other player receives the output of an oblivious transfer of this string and the erasures are independent for the different players.*

Lemma 20 *An authenticated oblivious broadcast can realize a GBC.*

Proof: Alice sends, as a commitment, k bit strings of length m (k, m are security parameters which are polynomial in n) with parity b. Then the knowledge all other players have about b is negligible in m. Because the probability that a bit is received by at least one player is $1 - 1/2^n$ and the probability that all players together have knowledge about all m is $(1 - 1/2^n)^m$ which is negligible in m. If k strings are sent the probability remains negligible as k and m are polynomial in n.

If Alice wanted to change the bit she committed to she has to change k bits. The probability that any single player does not detect this change is negligible in k. □

Lemma 21 *An authenticated oblivious broadcast can realize UOT.*

Proof: Alice creates a GBC and Bob publishes positions of two substrings of the strings Alice sent over the oblivious broadcast. One substring where he knows all the bits and one substring where he knows nothing. The substrings must have approximately the same length.

Alice publishes the bits of one of the substrings. Then Bob either learnt nothing new or he knows the bit Alice is committed to. We have realized UOT if we can show that no other player learns the bit Alice is committed to by the information published by Alice. But as the substrings published are statistically independent

of what the other players received this information just changes the probability of receiving a bit for each player. This change of probability can be coped with an suitable choice of the security parameters used in Lemma 20. □

VII. MAIN RESULT

Summarizing all of the above we can state:

Theorem 22 *The primitive of anonymous oblivious transfer is cryptographically strictly more powerful than oblivious transfer. It can realize all multi party protocols with a security and robustness which is independent from assumptions about possible collusions of cheaters or disruptors.*

Anonymous oblivious transfer can be realized by an authenticated oblivious broadcast channel or by a quantum protocol if no two possible collusions cover the set of players.

VIII. FUTURE WORK

An interesting question is if a noisy broadcast channel is of the same power as AOT. This seems to be clear for small sets of players, but if the number of players grow large the difference between the error probabilities possible for different collusions becomes large, too. If all players collude against the sender the probability of error is much lower as if all players collude against the receiver. To kope with this problem will be an interesting direction of future research.

There probably are many other primitives of a cryptographic power equivalent to AOT. This has to be investigated to maybe find primitives which can be realized more easily or more efficiently (compare [7]).

[1] Anonymous. Multi party protocols with oblivious transfer. The manuscript can be obtained via the authors of this paper, October 2000.

[2] D. Beaver and S. Goldwasser. Multiparty computations with faulty majority. In *Proceedings of the 30^{th} FOCS*, pages 468–473. IEEE, 1989.

[3] D. Chaum and S. Roijakkers. Unconditionally secure digital signatures. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology: Crypto '90*, volume 537 of *LNCS*, pages 206–215. Springer-Verlag, Berlin, 1990.

[4] R. Cleve. Controlled gradual disclosure schemes for random bits and their applications. In *Advances in Cryptology: Crypto '89*, pages 573–590, Berlin, 1989. Springer-Verlag.

[5] C. Crepeau. Efficient cryptographic protocols based on noisy channels. In *Advances in Cryptography: Eurocrypt 97*, Lecture Notes in Computer Science. Springer Verlag, 1997.

[6] C. Crepeau, J. van de Graaf, and A. Tapp. Committed oblivious transfer and private multi-party computations. In *Advances in Cryptology: Proceedings of Crypto '95*, pages 110–123. Springer, 1995.

[7] M. Fitzi, J. Garay, U. Maurer, and R. Ostrovsky. Oblivious cast and multi party protocols. Rump session of Crypto 2000, August 2000.

[8] S. Goldwasser and L. Levin. Fair computation of general functions in presence of immoral majority. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology: Crypto '90*, volume 537 of *LNCS*, pages 77–93. Springer-Verlag, Berlin, 1990.

[9] J. Müller-Quade and H. Imai. Quantum cryptographic three party protocols. Los Alamos preprint quant-ph/0010111, October 2000.

[10] J. Müller-Quade and H. Imai. Temporary assumptions for quantum multi party protocols. Technical Report of ISEC 11 Technical Meeting, Tokyo, The paper can be obtained via the authors of this paper, 2000.

[11] B. Pfitzmann and A. Waidner. Unconditional byzantine agreement for any number of faulty processors. In *Proc. STACS'92*, volume 577 of *LNCS*, pages 339–350. Springer-Verlag, Berlin, 1992. This paper generalizes the result of [3].