VERSITA

TATRA MOUNTAINS
Mathematical Publications
DOI: 10.2478/tmmp-2013-0037
Tatra Mt. Math. Publ. **57** (2013), 85–99

ANOTHER TWIST IN THE DINING CRYPTOGRAPHERS' PROTOCOL

Mihály Bárász — Péter Ligeti — Krisztina Lója –
– László Mérai — Dániel A. Nagy

ABSTRACT. In this paper, we explore the Dining Cryptographers' protocol over a cyclic group with a one-way homomorphic image, using a boardroom voting protocol as an illustration of its desirable security properties. In particular, we address the problem of anonymous disruption, which is one of the main disadvantages of DC over more usual groups like binary vectors.

1. Introduction

Anonymous broadcast is an immensely useful primitive for a variety of security-sensitive applications such as voting or auctions. For groups of limited size, D. Chaum proposed the so-called Dining Cryptographers' Protocol [3] (denoted, henceforth, by DC) which allows one participant to broadcast a message anonymously. In this protocol, every participant broadcasts (non-anonymously) a message; the anonymous broadcast is calculated as a sum of all them. A natural extension of this protocol is letting participants allocate random, unique and secret slots in a sequence of messages and have each broadcast a message anonymously in their own slot.

Because of the strong anonymity properties of DC, one of the most problematic attacks against it is *anonymous jamming*, which lets attackers prevent others from receiving the broadcasts, while still receiving them themselves. While legitimate broadcasters can *detect* that their message has been jammed, employing *reactive* security measures is difficult, because proving it would typically involve

2010 Mathematics Subject Classification: 94A60, 68P25.
Keywords: anonymous broadcast, boadroom voting, dining cryptographers.
This research has been partially supported by the Lendület program of the Hungarian Academy of Sciences. The second author also acknowledges the support of the grant OTKA PD-100712. The fourth author also acknowledges the support of the grant OTKA K-100291.

revealing sensitive information, undermining the anonymity of the broadcaster.

Another issue of DC is the unfair advantage that later broadcasters enjoy by knowing the information from previous broadcasts and making their broadcast dependent on them. This can be addressed by requiring that every participant broadcast their message for all slots in one turn. This, however, still leaves an unfair advantage with the last participant that learns all other broadcasts before deciding on his own.

The proposed scheme employs reactive defensive measures against both kinds of attack, by making it possible to catch cheaters without forcing honest participants to reveal sensitive information.

The main idea is doing three rounds of DC (over three different algebraic structures): a *slot reservation* round, a *commitment round* and an actual *broadcast round*. During reservation, participants agree on unique slots, learning only the position of their own slot. During commitment, each participant commits to the message, they are going to broadcast in the last round, without revealing the message itself. Any protocol violation in these two rounds can be investigated by having honest participants reveal the necessary information to prove their honesty; learning the intended broadcast message would be still computationally infeasible. Thus, protocol violators (those that cannot prove that they adhered to the protocol) can be punished. If we allow for colluding violators, at least one of them can be identified and punished. Also, these two rounds can be repeated, if unsuccessful.

By the third and final broadcast round, all participants have already committed to what they are going to broadcast in such a way that it is guaranteed that if they do so, the protocol will be successfully executed, without jamming any participant's broadcast. Also, at the time of commitment, none of the participants knows the actual message of other participants and thus cannot make their message dependent on those. Hence, no unfair advantage.

In the third round, each message can be immediately verified by any participant and/or a third party whether or not it conforms to the commitment. If it does not, the cheating participant is immediately identified.

This way, we constructed a cryptographic primitive that is most similar to collecting sealed notes in a ballot box.

1.1. Related work

In the original paper [3] Chaum describes in detail a protocol for anonymously broadcasting a single bit and generalizes it to sequences of bits. The basic protocol is described as follows in the introduction section of [3]: "*Each participant has a secret key bit in common with, say, every other participant. Each participant outputs the sum, modulo two, of all the key bits he shares, and if he* *wishes to transmit, he inverts his output. If no participant transmits, the modulo two sum of the outputs must be zero, since every key bit enters exactly twice; if one participant transmits, the sum must be one.*"

One of the biggest deficiencies of the original DC protocol originates exactly from its perfect anonymity. Namely, it can be disrupted by a malicious participant in a way that he learns the message(s) but no other participant does. And he can do this remaining perfectly anonymous. More precisely, it is very hard if at all possible, to determine the identity of the disruptive party while preserving the anonymity of the honest parties.

In his original paper [3], Chaum recognizes the problem and discusses it in detail in Section 2.5. He suggests using "traps" combined with slot reservation techniques to avoid this weakness at the cost of multiple broadcast rounds. Waidner and Pfitzmann [11] point out some weaknesses of these ideas and develop improved protocols.

A different approach to the problem is the use of zero-knowledge proofs instead of traps. Von Ahn et al. [1] construct a protocol with constant broadcast rounds by dividing the participants into groups and use secret sharing. The main drawback of this method is the high communicational and computational cost, especially in the presence of malicious participants.

Golle and Juels [7] propose two improved DC-variants with only two broadcast rounds. These protocols use bilinear maps and proofs of correctness as well.

2. The proposed protocol

Originally, Chaum proposed his DC protocol with bit strings and XOR operation, that is in our notion in group $G=\mathbb{Z}_2^K$. While he mentions that it could also be used in other communicative groups, he did not see any advantage in doing so.

Our idea, is to ameliorate exactly this deficiency. As it turns out, using other groups can help. In detail, by performing the DC protocol first in a one-way homomorphic image we can make every participant commit to some non-disruptive message (and deal with disruptions without loosing anonymity). After that, if an attacker still chooses to perform this attack with the plain messages, he is instantly identified.

The actual protocol begins with a slot reservation round, which is repeated until it is successful. Slot reservation is a standard Chaumian DC over binary vectors of appropriately chosen length K, where the message of each participant consists of $K-1$ zeroes and one set bit. The reservation round is successful, if and only if the number of ones in the summary vector equals the number of participants and all participants recognize their own one in the vector. The (zero-based) index of the reserved slot is the number of ones to the left of that of the participant in question.

Next comes the commitment round; DC performed in a cyclic group where the index (a.k.a. discrete log) problem is difficult. The actual message to which participants commit is the index, but it is unfeasible to calculate from the commitment messages (which are their homomorphic images). In this round, any participant can protest the result if his slot appears to be jammed and ask all participants to reveal all the keys (in the homomorphic image space and, if needed, from the successful slot reservation round). In case of actual protocol violation, at least one violator will get caught, while if there is no protocol violation, the one who raised the false alarm can be punished.

If the commitment round goes without irregularities, another round of DC is performed, in this case in the index space of the aforementioned cyclic group. Every message, as well as their sums, must be the indices of the corresponding values from the previous round. The summary message after this round will have the contribution of exactly one participant in each slot, which, in turn, are assigned in random order, analogously to the opening of a physical ballot box.

3. An application: a boardroom voting protocol based on DC-net

Here we present a voting system as one possible application of the above protocol and suggest an implementation.

Electronic voting systems which are designed for a limited number of participants are usually called *boardroom voting* systems in the literature.

The complexity of such systems could be measured by the number of required modular multiplications and exponentiations a participant must perform. The first published boardroom voting system considered by us is due to Kiayias and Yung [10], which uses $O(n^2)$ modular exponentiations, when n is the number of participants. Later Damgård and Jurik [5] proposed a bit similar scheme based on the Decisional Composite Residuosity assumption using quadratic number of exponentiations as well.

The protocol of Groth [9] requires a slightly smaller number of steps, the protocol uses $O(n)$ modular exponentiations in the registration phase and also in the verification of the ballots. Similarly to the protocol of Kiayias and Yung [10], the security of the system is based on the Decisional Diffie-Hellman assumption.

As the above-mentioned boardroom voting systems, our scheme also works in the random oracle model and assume an authenticated broadcast channel. In addition, the security of our protocol based on the Computational Diffie-Hellman assumption. The protocol uses a linear number of exponentiations only even in the presence of malicious participants.

3.1. Protocols used

Dining Cryptographers' protocol

For a given abelian group G and private data $t_i \in G$ belonging to the ith participant, the sum of these $t_1+\cdots+t_n$ can be computed by using the protocol without revealing any additional information about the individual data.

Let the ith participant's secret key be a_i and the public key be g^{a_i} $(i = 1, \dots, n)$, $f\colon \{0,1\}^* \to G$ be a collision free one-way function and ℓ a counter which increases in each round.

1. The participants compute their pairwise shared secret keys with the Diffie--Helmann protocol. The common secret key of pair (i, j) is $g^{a_i a_j}$.
2. One of the participants broadcast the value of the counter ℓ.
3. By using the shared keys the participants compute the common round keys $r_{i,j} \in F$ wherewith they encrypt their secret. The ith participant's keys are
$$r_{i,j} = \operatorname{sgn}(i-j) f(g^{a_i a_j} \| \ell), \qquad i \neq j.$$
4. Every participant i computes the encryption of t_i
$$S_i = \sum_{j\colon\, i\neq j} r_{i,j} + t_i\,.$$
5. Every participant publishes S_i. Since the sum of the round keys is zero, the sum of the secrets t_i is the sum of the published values
$$\sum_{i=1}^{n} S_i = \sum_{i=1}^{n} \left(\sum_{i\neq j} r_{i,j} + t_i \right) = \sum_{i=1}^{n} t_i\,.$$

Next, we will denote this protocol by $DC[t_1 + \cdots + t_n]$.

Chaum-Pedersen protocol

By using this generalized proof of knowledge protocol it is possible to provide proof of knowledge of x for given g, h and u, v such that the following equations hold:
$$g^x = u, \qquad h^x = v$$
without revealing any additional information about x.

It can be proven in the following way:

1. The prover chooses a random value w and computes $s = g^w$ and $t = h^w$.
2. The prover computes the following commitment $c = \mathcal{H}(g\|h\|u\|v\|s\|t)$ where $\mathcal{H}$ is a collision free hash function, and the value $r = w + cx$.
3. The certification is the (s, t, r, c) 4-tuple.
4. The certification is right if the following equations hold
$$g^r = su^c, \qquad h^r = tv^c.$$

This protocol will be denoted by $CP[x\colon g^x = u;\ h^x = v]$.

3.2. Security definitions

The security requirements we want the voting scheme to satisfy arise mostly from the boardroom voting literature (see [9], [10]), our system fulfill however some requirements of large-scale elections as well (for further voting system requirements see Gerck [8]). The desired requirements are the following:

1. **Perfect ballot secrecy**: this requirement ensures that any nontrivial knowledge about the partial tally of the ballots of any set of voters is only computable by the coalition of all the remaining voters.
2. **Self-tallying**: all participants and third parties are able to compute the result after the voting procedure.
3. **Universal verifiability**: every voter and outsider can be convinced that all votes have been counted properly in the final tally.
4. **Fairness**: nobody can modify the final tally even with some knowledge about a partial tally. (Note that this is a slight relaxation of the ordinary Fairness requirement claiming that nobody has knowledge about a partial tally before the end of the voting.)
5. **Every voter can vote exactly once**: undervotes are forbidden as well as overvotes. Because of the short period of time between registration and the voting round, we can assume that if a participant takes part in the first, s/he intends to take part in the second. It is, however, possible to allow for null-votes for those, who do register but do not intend to submit a vote.
6. **Opportunity to keep the transcript**: this is optional. Recording all the communication results in a transcript that would convince a third party that the voting was regular.
7. **Technology independent**: participants only need to trust the protocol itself and its implementation in their own device.
8. **Open source, open code**: the security of the system must not rely on the secrecy of the algorithm or the source code of the used programs. Only designated secrets (keys) and, of course, the votes must be kept secret.

3.3. The Voting protocol

The proposed protocol consists of four stages. The first one is *Registration*, when participants agree on shared secrets. The next one is *Slot reservation*, when the participant reserves his own slot using for broadcast communications during the next stages. The third is *Voting*, when participants broadcast their ballots in two steps, first, they broadcast a homomorphic function thereof to prevent cheating, then they broadcast the ballot itself. The last is *Investigation*, which is applied only in case of irregularities, when cheaters can be detected and disqualified by the honest voters.

Henceforth, we use the following notation: let n denote the number of participants, let p be a large enough prime and q a prime such that $q|p-1$ holds. Let

$$\mathbb{Z}_q \cong G \leq \mathbb{Z}_p^*$$

such that the Computational Diffie-Hellman assumption holds in G and g be a generator of G. Let us mention that in our case G need not satisfy all the usual conditions of Schnorr-groups. Note that q and p could be much smaller than usual (q about 80 bits and p about 600–800 bits, as we do not need long time secrecy, see Section 4).

Let $\mathcal{D}_i(x)$ denote the digital signature of the ith participant over message x. Let

$$f\colon \{0,1\}^* \to \mathbb{Z}_q^n$$

be an arbitrary one-way function and $\mathcal{H}$ be an arbitrary hash-function which is used for digital signatures.

Registration:

1. Every participant chooses a secret key and a corresponding public key (the ith participant's keys are a_i, g^{a_i} resp.).
2. The participants compute their pairwise shared keys: let i, j pair's common key be $g^{a_i a_j}$.

Slot reservation:

1. Every participant chooses a binary vector which has only one non-zero coordinate, the ith participant's vector is $e_i \in \{0,1\}^K$, where K is about $n^2/2$, see Remark 1.
2. The participants broadcast the vectors: $DC[e_1 + e_2 + \cdots + e_n]$. (So we perform the DC protocol in $\mathbb{Z}_2^K$ here.)
3. There are three cases:
   - If the number of 1's in vector $e_1 + e_2 + \cdots + e_n$ is less than n, then there were collisions (some of the participants choose the same vector). In this case, the participants return to the step 1.

- If the number of 1's in vector $e_1 + e_2 + \dots + e_n$ is n and every participant can find "his own 1" in the result, then the *Slot reservation* was successful. Suppose that the d_ith 1 derive from the ith participant. Then let the ith participant's slot be the d_ith.
- If the number of 1's in vector $e_1 + e_2 + \dots + e_n$ is more than n or any of the participants could not find a 1 in his position, then someone has violated the protocol. In this case we can apply the *Investigation* stage.

Remark 1. It is practical, to choose K, such that it minimizes the expected value of communicated bits. The optimal K is about $n^2/2$. Furthermore, if every participant chooses a random slot, then the expected number of broadcast rounds needed for a collision-free slot reservation is 3. For more details, see Appendix.

Voting:

1. Suppose the ith participant's vote is $k_i \in \{0,1\}^k$ (here we assume that votes are encoded in k bits). His/her message is m_i such that
$$m_i = k_i \| r_i,$$
where r_i is a random string with fix length such that m_i is less than q.
2. The participants compute round keys. One of the participants broadcasts the value of the counter ℓ. The ith participant's keys are
$$s_{i,j} = \operatorname{sgn}(i-j) f(g^{a_i a_j} \| \ell) \in \mathbb{Z}_q^n, \qquad i \neq j \tag{1}$$
represented by the following n-tuple
$$s_{i,j} = \big(s_{i,j}(1), s_{i,j}(2), \dots, s_{i,j}(n)\big)$$
(here $s_{i,j}(l) \in \mathbb{Z}_q$).
3. The participants compute commitments. The ith participant's commitment is
$$F_i = \big(F_i(1), F_i(2), \dots, F_i(n)\big),$$
here
$$F_i(d_i) = g^{\sum_{i \neq j} s_{i,j}(d_i) + m_i} \tag{2}$$
and
$$F_i(t) = g^{\sum_{i \neq j} s_{i,j}(t)} \qquad \text{if } t \neq d_i. \tag{3}$$
4. The participants publish commitments and check them. The ith participant checks whether the commitments satisfy the following equation:
$$\prod_{j=1}^{n} F_j(d_i) = g^{m_i}.$$

If one of the equation does not hold, then someone has violated the protocol. In this case we perform the *Investigation* stage.
Note, this step corresponds to a DC protocol in the G^n group.
In the case when a transcript is required, the participants sign the commitments. The ith part of transcript is

$$\mathcal{D}_i\big(\mathcal{H}(F_1\|F_2\|\dots\|F_n)\big).$$

5. When every participant accepted the commitments, they publish the exponents. The exponent vector E_i of the ith participant consists of n parts:

$$E_i(d_i) = \sum_{i\neq j} s_{i,j}(d_i) + m_i \tag{4}$$

and

$$E_i(t) = \sum_{i\neq j} s_{i,j}(t) \qquad \text{if } t \neq d_i. \tag{5}$$

Note, this step corresponds to a DC protocol in the $\mathbb{Z}_q^n$ group.

6. The participants verify the exponent. The ith participant is honest if the following equations hold:

$$g^{E_i(t)} = F_i(t), \qquad t = 1, 2, \dots, n.$$

The sum of vectors E_i is the ballot of participants.

3.4. Investigation

In this section, we show how attackers can be identified.

Slot reservation: Someone violated the protocol, i.e., the vector of the attacker consists of more than one non-zero coordinates.

1. Every participant publishes all of the round keys of slot reservation (the keys of the ith participant are $r_{i,j}$, $j \neq i$) and check them, i.e., they verify whether the following equations hold:

$$r_{i,j} = -r_{j,i}, \qquad i, j = 1, 2, \dots, n.$$

There are two cases:
   - the equations hold, then the vectors e_i can be computed and the cheater can be identified.
   - there is a pair (i, j), where $r_{i,j} \neq -r_{j,i}$, then either the ith or the jth is adversary.

2. Now, the participants i, j can prove their honesty in the following way. Let their shared round keys be r_i, r_j ($r_i \neq -r_j$), the value of the shared key $g^{a_i a_j}$ is u_i, u_j (as reported by i and j, respectively).

They publish the value of the shared key and its verification:

– ith participant publishes

$$u_i \quad \text{and} \quad CP\big[a_i' : g^{a_i'} = g^{a_i}(g^{a_j})^{a_i'} = u_i\big];$$

– jth participant publishes

$$u_j \quad \text{and} \quad CP\big[a_j' : g^{a_j'} = g^{a_j}, (g^{a_i})^{a_j'} = u_j\big].$$

It can be verified that

$$r_i = f(u_i \| \ell) \qquad \text{or} \qquad r_j = f(u_j \| \ell).$$

The equation of the cheater does not hold.

Voting: Suppose that the ith participant does not accept the commitment in the step 4, see Subsection 3.3, i.e.,

$$\prod_{j=1}^{n} F_j(d_i) \neq g^{m_i}.$$

In this case, the participants publish their shared round keys (the keys of the jth participant are $s_{j,l}$). If there is a pair of keys (for example j, l) such that $s_{j,l} \neq -s_{l,j}$, then the cheater can be identified using the previous method. If the keys are correct (i.e., $s_{j,l} = -s_{l,j}$ $j, l = 1, 2, \ldots, n$), then it can be deduced who used the d_ith slot. If this slot is used by only the ith participant, then he is cheater, else there is a set of participants which includes only one honest participant. Then they publish the shared round keys of the Slot reservation stage. Using these keys, it can be found out who reserved slot d_i.

3.5. Security analysis

In the following we prove that the proposed protocol is secure under the random oracle assumption [2] and the Computational Diffie-Hellman assumption [6].

3.5.1. Perfect Ballot Secrecy

Let us first note that if Perfect Ballot Secrecy does not hold then there exists a subset $\mathcal{C}$ of the participants, a partition $\mathcal{B}_1 \cup \mathcal{B}_2 = \bar{\mathcal{C}}$ of the remaining participants and a polynomial algorithm $\mathcal{A}$ such that coalition $\mathcal{C}$ can compute the partial tally of $\mathcal{B}_1$ and $\mathcal{B}_2$ by using the algorithm $\mathcal{A}$. Let us denote this event by PBS(n), where n is the number of the participants. Hence it is enough to prove the following

Theorem 1. *Suppose that every secret key a_i is chosen independently from a uniform distribution and there exists some ε such that $P\big(\mathrm{PBS}(n)\big) > \varepsilon$. Then there exists a polynomial p such that the probability that there are a, b such that one can compute g^{ab} knowing g^a and g^b is greater than $\varepsilon/p(n)$.*

Proof. Let the public keys of two participants (say A and B) be g^a and g^b and let the private keys x_i ($i > 2$) for every other participant be chosen randomly. Finally, compute the corresponding public keys g^{x_i}.

Let c denote the probability of the event that A and B are in different $\mathcal{B}_i$s. Note that $c = 1/p(n)$ for a suitable polynomial p. Thus, we can assume that A and B are in different $\mathcal{B}_i$s. The probability of this is greater than εc.

Now all the private keys (except the private keys of A and B) are given to $\mathcal{A}$ as input. By definition, $\mathcal{A}$ can compute the partial tally of $\mathcal{B}_1$

$$\sum_{i \in \mathcal{B}_1} m_i \, .$$

Hence $\mathcal{C}$ can compute m_A using $\mathcal{A}$ since the ballots of all further participants are known. From the secret keys x_i ($i > 2$) and the public key g^a, $\mathcal{C}$ can compute all round keys $s_{A,j}$ of participant A except $s_{A,B}$ from (1). Then with the help of m_A, $E_A(d_A)$ and these round keys, coalition $\mathcal{C}$ can compute the round key $s_{A,B} = f(g^{ab} \| \ell)$ from (4) (knowing public keys g^a, g^b and counter ℓ). As f is modeled as a random oracle, $\mathcal{C}$ can do it only in the case when it can compute the value of g^{ab} with probability greater than εc. □

3.5.2. Every voter can vote exactly once

Since the number of the slots equals the number of participants, if an attacker can vote more than once then he can only do it on behalf of someone else. However, if somebody's vote does not appear in the commitment stage, then by using *Investigation*, only a honest participant can prove that he is entitled to use that slot.

3.5.3. Fairness

As in *Perfect Ballot Secrecy* the hardness of the discrete logarithm problem ensures that participants have no knowledge about the partial tally in the Commitment stage. However, after this stage participants cannot modify their ballots. If all participants publish their exponents, then the vote can be completed. Otherwise, participants who do not publish their exponents can be considered malicious.

3.6. Complexity

In the following we summarize the main computational and communicational characteristics of the proposed boardroom voting protocol. In Table 1 we consider two cases: when none of the participants violate the protocol (optimistic case) and when malicious participants are present (pessimistic case). Let us recall that n is the number of participants and p is a sufficiently large prime, i.e., $\log p \approx 800$.

Table 1. The complexity of the voting protocol.

	Optimistic	Pessimistic
Nr. of exponentiations	$2n$	$O(n)$
Nr. of one-way functions	$4n-2$	$O(n)$
Nr. of communicated bits	$2n\log p + O(n^2)$	$O(n\log p) + O(n^2)$

4. Conclusions

We have successfully addressed a major drawback of conventional DC protocols, where broadcast anonymity results in jammer anonymity and, hence, jammer impunity.

The boardroom voting protocol in our example is quite comparable to other proposed boardroom voting schemes (e.g., [9], [10]) in terms of security properties (self-tallying, trust limited to own device, third-party verifiable, perfect ballot secrecy, fairness, etc.) and the resources utilized to achieve them. In particular, we do not require a trusted third party.

Although the large number of exponentiations used in this protocol might seem as prohibitively expensive, it is important to emphasize that most of them are employed to prevent attackers from learning actual message contents from committments. The time available for such an attack is very limited due to the rapid succession of the committment round and the actual broadcast round, when all indices are revealed anyway. The lack of the long-time secrecy requirement allows for much smaller and hence cheaper groups for homomorphic transformation than the ones typically used in asymmetric cryptography. In fact, the proposed protocol can be implemented in low-end mobile phones if G_2 is chosen to be the subgroup generated by 2 of a multiplicative group of a Galois field of the moduli of some Sophie Germain prime of a few hundred bits.

The proposed scheme can be utilized for purposes other than voting. We chose voting as our example, because the desirable security properties of voting are better understood and researched than other possible applications of fair anonymous broadcast with exactly one message per participant.

Thus, our proposed twist on DC results in a cryptographic primitive that is functionally equivalent to a (small) ballot box. In addition to voting, it can also be applied to sealed bid auctions and other similar problems.

REFERENCES

[1] VON AHN, L.—BORTZ, A.—HOPPER, N. J.: *k-anonymous message transmission,* in: Proc. of the 10th ACM Conf. on Computer and Commun. Security––CCS '03 (S. Jajodia, ed.), Washington, DC, USA, 2003, ACM, New York, USA, 2003, pp. 122–130.

[2] BELLARE, M.—ROGAWAY, P.: *Random oracles are practical: a paradigm for designing efficient protocols,* in: Proc. of the 1st ACM Conf. on Comput. and Commun. Security—CCS '93 (D. E. Denning et al., eds.), Fairfax, Virginia, USA, 1993, ACM, New York, NY, USA, pp. 62–73.

[3] CHAUM, D.: *The dining cryptographers problem: unconditional sender and recipient untraceability,* J. Cryptology **1** (1988), 65–75.

[4] CHAUM, D.—PEDERSEN, T. P.: *Wallet databases with observers,* in: Advances in Cryptology—CRYPTO '92, Proc. of the 12th Annual Internat. Cryptology Conf. on Advances in Cryptology (E. F. Brickell, ed.), Santa Barbara, California, USA, 1992, Lecture Notes in Comput. Sci., Vol. 740, Springer, Berlin, 1993, pp. 89–105.

[5] DAMGÅRD, I. B.—JURIK, M. J.: *A length-flexible threshold cryptosystem with applications,* in: Proc. of 8th Australasian Conf. Inform. Security and Privacy—ACISP '03 (R. Safavi-Naini and J. Seberry, eds.), Wollongong, Australia, 2003, Lecture Notes in Comput. Sci., Vol. 2727, Springer, Berlin, 2003, pp. 350–364.

[6] DIFFIE, W.—HELLMAN, M. E.: *New directions in cryptography,* IEEE Trans. Inform. Theory **22** (1976), 644–654.

[7] GOLLE, P.—JUELS, A.: *Dining cryptographers revisited,* in: Advances in Cryptology––Eurocrypt '04, Internat. Conf. on the Theory and Appl. of Cryptogr. Techniques (Ch. Cachin and J. L. Camenisch, eds.), Interlaken, Switzerland, 2004, Lecture Notes in Comput. Sci., Vol. 3027, Springer, Berlin, 2004, pp. 456–473.

[8] GERCK, E.—NEFF, C. A.—RIVEST, R. L.—RUBIN, A. D.—YUNG, M.: *The business of electronic voting, in financial cryptography,* in: Financial Cryptography—FC '01, 5th Internat. Conf. (P. F. Syverson, ed.), Grand Cayman, British West Indies, 2001, Lecture Notes in Comput. Sci., Vol. 2339, Springer, Berlin, 2001, pp. 234–259.

[9] GROTH, J.: *Efficient maximal privacy in boardroom voting and anonymous broadcast,* in: Financial Cryptography—FC '04, 8th Internat. Conf. (A. Juels, ed.), Key West, FL, USA, 2004, Lecture Notes in Comput. Sci., Vol. 3110, Springer, Berlin, 2004, pp. 90–104.

[10] KIAYIAS, A.—YUNG, M.: *Self-tallying elections and Perfect Ballot Secrecy,* in: Public Key Cryptography—PKC '01, 5th Internat. Workshop on Practice and Theory in Public Key Cryptosystems (D. Naccache and P. Paillier, eds.), Paris, France, 2002, Lecture Notes in Compu. Sci., Vol. 2274, Springer, Berlin, 2002, pp. 141–158.

[11] WAIDNER, M.—PFITZMANN, B.: *The dining cryptographers in the disco—underconditional sender and recipient untraceability with computationally secure serviceability (abstract),* in: Advances in Cryptology—EUROCRYPT '89, Workshop on the Theory and Appl. of Cryptographic Techniques (J.-J. Quisquater, J. Vandewalle, eds.), Houthalen, Belgium, 1989, Lecture Notes in Comput. Sci., Vol. 434, Springer, Berlin, 1989, p. 690.

Appendix

In slot reservation, our goal is to minimize the expected amount of necessary communication until each participant is satisfied that they have an exclusive, randomly selected slot for their ballot broadcast.

Let us assume, that we have n participants and K slots. There exist K^n possibilities of putting the messages into the network. From these in $n!\binom{K}{n}$ cases there is no collision. So, the probability of no collision is:

$$P(n,K) = \frac{n! \cdot \binom{K}{n}}{K^n}.$$

Now we can compute for every n the optimal K which minimizes $\frac{K}{P(n,K)}$.

Table 2. The optimal parameters for collision-free slot reservation.

n	K_{opt}	$\lceil n^2/2 \rceil$	$P(n, K_{opt})$	$l_{opt} = \frac{K_{opt}}{P(n,K_{opt})}$
2	2	2	0.5	4
3	5	5	0.48	10.4
4	8	8	0.410	19.50
5	13	13	0.416	31.25
6	19	18	0.415	45.75
7	25	25	0.397	62.98
8	33	32	0.398	82.91
9	42	41	0.398	105.56
10	51	50	0.389	130.93
20	203	200	0.380	534.14
30	455	450	0.376	1209.18
40	806	800	0.374	2156.05
50	1258	1250	0.372	3374.74
100	5018	5000	0.370	13545.65
500	125083	125000	0.368	339558.25

In Table 2 We can see that the optimal K is very close to $n^2/2$. If we use $\lceil n^2/2 \rceil$ slots, then the expected number of bits communicated is just a bit more than with the optimal K, if the number of the participants is less than 500 (and with 500 participants this scheme is already impractical). With this choice the probability that more that one round is necessary in the slot reservation phase is about 40 %, which means that most of the time no more than three rounds will be needed.

Received September 18, 2013

Mihály Bárász
ELTECRYPT Research Group
E-mail: klao@cs.elte.hu

Péter Ligeti
Department of Computeralgebra
and ELTECRYPT
E-mail: turul@cs.elte.hu

Krisztina Lója
ELTECRYPT Research Group
E-mail: loja@cs.elte.hu

László Mérai
Department of Computeralgebra
and ELTECRYPT
E-mail: merai@cs.elte.hu

Dániel A. Nagy
ELTECRYPT Research Group
E-mail: nagydani@cs.elte.hu

Eötvös Loránd University
Pázmány P. sétány 1/C
H–1117 Budapest
HUNGARY