The Power of Anonymous Veto in Public Discussion

Feng Hao[1] and Piotr Zieliński[2,⋆]

[1] Thales Information Systems Security, nCipher product line
[2] Google Inc.
feng.hao@thales-esecurity.com, piotrzielinski@google.com

Abstract. The Dining Cryptographers problem studies how to securely compute the boolean-OR function while preserving the privacy of each input bit. Since its first introduction by Chaum in 1988, it has attracted a number of solutions over the past twenty years.

In this paper, we propose an exceptionally efficient solution: Anonymous Veto Network (or AV-net). Our protocol is provably secure under the Decision Diffie-Hellman (DDH) and random oracle assumptions, and is better than past work in the following ways. It provides the strongest protection of each input's privacy against collusion attacks; it requires only two rounds of broadcast, fewer than any other solution; the computational load and bandwidth usage are the least among the available techniques; and the efficiency of our protocol is achieved without relying on any private channels or trusted third parties. Overall, the efficiency of our protocol seems as good as one may hope for.

Keywords: Dining Cryptographers problem; DC-net; anonymous veto; secure multiparty computation.

1 Introduction

In a galaxy far far away ...

During an open meeting, the Galactic Security Council must decide whether to invade an enemy planet. One delegate wishes to veto the measure, but worries that such a move might jeopardize the relations with some other member states. How can he veto the proposal without revealing his identity?

The above shows a picture of the council delegates – with mutual suspicion – discussing a decision in public. There are no private channels. The only way to communicate between each other is through public announcement; and during announcement, every word uttered or message sent can be traced back to its sender. There is no external help either, as trusted third parties do not exist.

⋆ The work was done when both authors were at the Computer Laboratory, University of Cambridge. This is an extended version of an earlier conference paper [1].

M.L. Gavrilova et al. (Eds.): Trans. on Comput. Sci. IV, LNCS 5430, pp. 41–52, 2009.

In essence, this problem requires a secure computation of the boolean-OR function, while preserving the privacy of each input bit. It was coined by Chaum as the Dining Cryptographers problem [5]; however, the "unconditional secrecy channels" assumed in [5] are no longer readily available in our case, which makes the delegate's task harder.

There have been a number of solutions in past work, ranging from circuit evaluation [20,10] and Dining Cryptographers Network (or DC-net) [5] proposed nearly twenty years ago, to several anonymous veto protocols [15,11,4] published in recent years. However, these techniques all have various limitations, as we discuss now.

The DC-net protocol was proposed by Chaum in 1988, and has long been considered a classic privacy-preserving technique [12]. This protocol has two stages. First, n participants set up pairwise shared secrets through secret channels. Next, each participant P_i broadcasts a one bit message a_i, which is the XOR of all the shared one-bit secrets that P_i holds if he has no message (i.e., no veto) to send, or the opposite bit otherwise. After the broadcast round, the sent message is decoded by all participants through computing the XOR of the broadcast bits. More details can be found in [5].

However, deploying DC-nets is hampered for several reasons. First, the "unconditional secrecy channels" are assumed in the protocol, but difficult to achieve in practice. This problem is further compounded by the rapid increase of the total number of such channels (i.e., $O(n^2)$) when there are more participants. Second, message collisions are problematic too. Even when all participants are honest, an even number of messages will still cancel each other out, forcing retransmissions. Third, a DC-net is vulnerable to malicious jamming. For example, the last participant P_n may send $\oplus_{i=1}^{n-1} a_i$, so that the final outcome will always be '0' (i.e., non-veto). Countermeasures include setting up "traps" to catch misbehaviors probabilistically, but make the system more complex (see [5, 12, 18]).

While a DC-net is "unconditionally secure", all the other solutions are built upon public key cryptography, and are thus computationally secure. These include the circuit evaluation technique [10] and several anonymous veto protocols [15,11,4]. A lack of efficiency is their common problem. We will explain this in more detail in Section 4.

Despite the problems in a DC-net, we still find it, among all the past solutions, most attractive for its simplicity and elegance. It combines all others' secret keys to encrypt data, but requires no secret keys to decrypt it. This idea is seminal, but undeservedly, has rarely been exploited in secure multiparty computations for the past twenty years.

By contrast, the mix-net protocol – a twin technique introduced by Chaum to protect anonymity – has been extensively studied and applied in the field [6]. It encrypts messages in multiple layers using public key cryptography, and usually requires a chain of proxy servers to perform secure decryption. In comparison, a DC-net is more lightweight; it sends anonymous data by a one-round broadcast and allows rapid decryption with no servers needed.

Our solution, Anonymous Veto Network (or AV-net), captures the essence of the original DC-net design [5] – it combines everyone else's public key to encrypt data, but requires no private keys to decrypt it. This, as we will show in Section 4, leads to the optimal efficiency of our protocol in many aspects. However, despite the similarity in the underlying design principles, the technical developments for the DC-net and AV-net protocols are completely different. In the following section, we will explain how an AV-net works.

2 Protocol

2.1 Model

We assume an authenticated public channel available for every participant. This assumption is made in all the past work in this line of research [5, 15, 11, 4]; in fact, an authenticated public channel is an essential requirement for general multi-party secure computations [10, 3]. This requirement basically ensures that the published votes come from the legitimate or registered voters; otherwise, voting would be meaningless. There are several ways to realize such a channel: by using physical means or a public bulletin board [15]. Apart from this basic requirement, we do not assume any secret channels or trusted third parties.

In the threat model, we consider two types of attackers: a passive one who merely eavesdrops on the communication, and an active one who takes part in the voting. Active attackers may collude in an effort to uncover others' votes or manipulate the voting result. The *full collusion* against a participant involves all the other participants in the network. Any anonymous veto protocol, by nature, cannot preserve the vetoer's anonymity under this circumstance. However, as explained in [5], it is practically impossible to have all participants – who mistrust each other – colluding against just one; there would be no point for that person to stay in the network. Hence, in this paper, we only consider *partial collusion*, which involves only some participants, but not all.

Under the threat model of partial collusion, an anonymous veto protocol should satisfy the following three requirements.

- *Veto Privacy* – If one vetoes, the rest of the participants cannot tell who has vetoed.
- *Veto Completeness* – If one vetoes, all participants accept the veto outcome.
- *Veto Soundness* – If the outcome is veto, all participants accept that someone has vetoed.

Clearly, a DC-net does not satisfy the second requirement, because of the collision problem. More requirements are defined in [15, 11] to reflect the trustworthiness of the third parties involved, but are not needed in our model.

2.2 Two-Round Broadcast

Let G denote a finite cyclic group of prime order q in which the Decision Diffie-Hellman (DDH) problem is intractable [2]. Let g be a generator in G. There

are n participants, and they all agree on (G, g). Each participant P_i selects a random value as the secret: $x_i \in_R \mathbb{Z}_q$.

Round 1. *Every participant P_i publishes g^{x_i} and a knowledge proof for x_i.*

When this round finishes, each participant P_i computes

$$g^{y_i} = \prod_{j=1}^{i-1} g^{x_j} \Big/ \prod_{j=i+1}^{n} g^{x_j}$$

Round 2. *Every participant publishes a value $g^{c_i y_i}$ and a knowledge proof for c_i, where c_i is either x_i or a random value $r_i \in_R \mathbb{Z}_q$, depending on whether participant P_i vetoes or not.*

$$g^{c_i y_i} = \begin{cases} g^{r_i y_i} & \text{if } P_i \text{ sends '1' (veto),} \\ g^{x_i y_i} & \text{if } P_i \text{ sends '0' (no veto).} \end{cases}$$

To check the final message, each participant computes $\prod_i g^{c_i y_i}$. If no one vetoes, we have $\prod_i g^{c_i y_i} = \prod_i g^{x_i y_i} = 1$. This is because $\sum_i x_i y_i = 0$ (Proposition 1). Hence, $\prod_i g^{x_i y_i} = g^{\sum_i x_i y_i} = 1$.

On the other hand, if one or more participants send the message '1', we have $\prod_i g^{c_i y_i} \neq 1$. Thus, the one-bit message has been sent anonymously.

Proposition 1 (Soundness). *For the x_i and y_i defined in an AV-net, $\sum_i x_i y_i = 0$.*

Proof. By definition $y_i = \sum_{j<i} x_j - \sum_{j>i} x_j$, hence

$$\begin{aligned} \sum_i x_i y_i &= \sum_i \sum_{j<i} x_i x_j - \sum_i \sum_{j>i} x_i x_j \\ &= \sum_{j<i} \sum x_i x_j - \sum_{i<j} \sum x_i x_j \\ &= \sum_{j<i} \sum x_i x_j - \sum_{j<i} \sum x_j x_i \\ &= 0. \end{aligned}$$

Table 1 illustrates this equality in a more intuitive way.

The above proposition shows that if no one has vetoed, the outcome will be non-veto. Equivalently, if the outcome is veto, someone must have vetoed. This shows that the protocol fulfills the "veto soundness" requirement defined in Section 2.1.

In the protocol, senders must demonstrate their knowledge of the discrete logarithms, namely the secrets x_i and c_i, without revealing them. This can be realized by using a Zero-Knowledge Proof (ZKP), a well-established primitive in cryptography [7, 8, 16, 9].

As an example, we could use Schnorr's signature, for it is non-interactive, and reveals nothing except the one bit information about the truth of the statement:

Table 1. A simple illustration of $\sum_{i=1}^{n} x_i y_i = 0$ for $n = 5$. The sum $\sum_{i=1}^{n} x_i \, (\sum_{j=1}^{i-1} x_j - \sum_{j=i+1}^{n} x_j)$ is the addition of all the cells, where $+$, $-$ represent the sign. They cancel each other out.

	x_1	x_2	x_3	x_4	x_5
x_1		$-$	$-$	$-$	$-$
x_2	$+$		$-$	$-$	$-$
x_3	$+$	$+$		$-$	$-$
x_4	$+$	$+$	$+$		$-$
x_5	$+$	$+$	$+$	$+$	

"the sender knows the discrete logarithm" [16]. Note that Schnorr's signature is provably secure under the random oracle model, so our scheme would also work in the random oracle model – that is requiring a secure hash hash function. Let H be such a secure hash function. To prove the knowledge of the exponent for g^{x_i}, one can send $\{g^v, r = v - x_i h\}$ where $v \in_R \mathbb{Z}_q$ and $h = H(g, g^v, g^{x_i}, i)$. This signature can be verified by anyone through checking whether g^v and $g^r g^{x_i h}$ are equal. Note that here the participant index i is unique and known to all. Adding i inside the hash function can effectively prevent a replay of this signature by other participants. Other ZKP techniques can be found in [9].

3 Security Analysis

Simplicity is the main goal in our protocol design. The construct of AV-net is quite straightforward, in fact much simpler than the related works [15, 11, 4]. In addition, we built the protocol upon the well-established technique such as Schnorr's signature whose zero-knowledge property has been well-understood. This greatly simplifies the task of security analysis. Therefore, in this section, we shall aim to provide intuitive (yet rigorous) security analysis without having to go through the lengthy formalism (though formal models can be instantiated).

In the AV-net construct, each participant sends out ephemeral public keys in the first round, and then encrypts his vote by combining the public keys in the second round. To breach the anonymity of a participant, an observer – anyone within the broadcast range – may try to uncover the one-bit message from the announced ciphertext. In the following, we will prove that, under the DDH assumption, the proposed cryptosystem achieves *semantic security* [13]. This is equivalent to showing that under the hard-problem assumption, ciphertext is indistinguishable to observers [13].

In an AV-net, the value of y_i is determined by the private keys of all participants except P_i. The following lemma shows its security property.

Lemma 2. *In an AV-net, y_i is a secret random value to attackers in partial collusion against the participant P_i.*

Proof. Consider the worst case where only P_k $(k \neq i)$ is not involved in the collusion. Hence x_k is uniformly distributed over $\mathbb{Z}_q$ and unknown to colluders.

The knowledge proofs required in the protocol show that all participants know their private keys. Since y_i is computed from x_j $(j \neq i, k)$ known to colluders plus (or minus) a random number x_k, y_i must be uniformly distributed over $\mathbb{Z}_q$. Colluders cannot learn y_i even in this worst case.

Theorem 3 (Privacy). *Under the Decision Diffie-Hellman assumption, attackers in partial collusion against P_i cannot distinguish the two ciphertexts $g^{x_i y_i}$ and $g^{r_i y_i}$.*

Proof. Besides the sent ciphertext, the data available to attackers concerning P_i include: g^{x_i}, g^{y_i} and Zero-Knowledge Proofs for the proof of the exponents x_i, y_i. The secret x_i is chosen randomly by P_i. Lemma 2 shows that y_i is a random value, unknown to the attacker. The ZKP only reveal one bit: whether the sender knows the discrete logarithm[1]; it is provable that it does not leak anything more than that [16]. Therefore, according to the Decision Diffie-Hellman assumption, one cannot distinguish between $g^{x_i y_i}$ and a random value in the group such as $g^{r_i y_i}$ [2].

The above theorem states that the individual published ciphertext does not leak any useful information. It is the multiplication of all ciphertexts that tells the outcome. For each participant, the learned information from the protocol is strictly confined to the multiplied result plus his own input. If a participant vetoes, the rest of the participants cannot track down the vetoer without full collusion. This shows that the protocol fulfills the "veto privacy" requirement defined in Section 2.1.

An anonymous veto protocol must resist jamming, to which a DC-net is vulnerable. Apparently, in the second round of broadcast, we need g^{y_i} be a generator for the group, so that participant P_i can produce a valid signature. Because the group G has prime order, any non-identity element is a generator. From Lemma 2, the value y_i is random over $\mathbb{Z}_q$ even in the face of active attacks (partial collusion). Thus, given that q is a large number, say 160-bit [2], the chance that $y_i \neq 0$ is overwhelming: $1 - 2^{-160}$. Only in the full collusion case can attackers manipulate $y_i = 0$, which then makes full collusion immediately evident. The resistance to jamming in an AV-net is formally proved below.

Theorem 4 (Completeness). *Under the Discrete Logarithm assumption, if P_i vetoes, provided that g^{y_i} is not the identity element in the group, P_i's veto cannot be suppressed.*

Proof. Assume P_i's veto can be suppressed, and we will show that one can then solve the Discrete Logarithm problem. Given g^{r_i}, where r_i is a random value, one can compute r_i by simulating the protocol with jamming: participant P_i announces $g^{r_i y_i} = (g^{r_i})^{y_i}$, but his veto is suppressed by others. The simulator generates all other secrets, except $c_i = r_i$. By definition we have $\prod g^{c_i y_i} = 1$.

[1] It should be noted that if we choose Schnorr's signature to realize ZKPs, we implicitly assume a random oracle (i.e., a secure hash function), since Schnorr's signature is provably secure under the random oracle model [16].

Table 2. Comparison to the past work

related work	pub year	rnd no	know proof	pvt ch	colli- sion	3rd pty	collu- sion	security reliance	system compl
GMW [10]	1987	3	$O(n)$	yes	no	no	half	trapdoor	$O(n^2)$
Chaum [5]	1988	2+	—	yes	yes	no	full	uncond	$O(n^2)$
KY [15]	2003	3	$O(n)$	no	no	yes	full	DDH	$O(n^2)$
Groth [11]	2004	$n+1$	2	no	no	yes	full	DDH	$O(n)$
Brandt [4]	2005	4	4	no	no	no	full	DDH	$O(n)$
AV-net	—	**2**	**2**	**no**	**no**	**no**	**full**	**DDH**	$\boldsymbol{O(n)}$

That is $g^{r_i y_i} = \prod_{j \neq i} g^{-c_j y_j}$. The knowledge proofs required in the protocol show that the simulator knows the values x_j $(1 \leq j \leq n)$ and c_j $(1 \leq j \leq n$ and $j \neq i)$. Also note that g^{y_i} is not an identity element, so $y_i \neq 0$. Hence, the simulator can easily compute $r_i = y_i^{-1} \sum_{j \neq i} -c_j y_j$, where $y_i = \sum_{j<i} x_j - \sum_{j>i} x_j$. With the obtained knowledge of the r_i value, the simulation is complete. Thus, one solves the discrete logarithm of g^{r_i} by simulating the protocol with jamming. This, however, contradicts the Discrete Logarithm assumption.

The above theorem states that jamming the protocol implies solving the Discrete Logarithm problem, which is believed to be intractable. In other words, the protocol ensures that when a participant vetoes, his veto message will be received by all. This makes the protocol fulfill the "veto completeness" requirement defined in Section 2.1.

Overall, the zero-knowledge proof, as a crypto primitive, is important in our security analysis. Without it, several attacks would be possible. If there were no knowledge proofs in the first round, participant P_n could manipulate the value of y_1 by announcing $1/\prod_{i=2}^{n-1} g^{x_i}$, so that $y_1 = 0$. Similarly, if there were no knowledge proofs in the second round, the last participant P_n could jam the protocol by announcing $1/\prod_{i=1}^{n-1} g^{c_i y_i}$. Hence, the zero-knowledge proof is the technique to make the protocol self-enforcing – ensuring that participants do perform the asymmetric operations (e.g., exponentiation) as stated, rather than give out random data. With the exception of the DC-net protocol, it is required in all the other solutions based on public key cryptography.

4 Efficiency

For the past twenty years, there have been a few techniques available to compute the boolean-OR function securely. They are summarized in Table 2.

Among all solutions, an AV-net stands out for its optimal efficiency in many aspects. First, it needs only two rounds, fewer than any others. In fact, two is the best round-efficiency achievable (see Appendix A). Second, it takes only a single exponentiation to encrypt data, no matter how many participants there are. Third, the size of the broadcast ciphertext $g^{c_i y_i}$ is only half of that using the standard ElGamal encryption (see [4]). It seems unlikely to be reduced further.

The AV-net protocol adopts the ZKP primitive, which may require additional computation in verification. The exact computational cost depends on the choice of the specific ZKP technique, whether the outcome is in doubt and the trust relationships between participants. It is also significant that since all communication is public in our protocol, any invalid ZKPs would present themselves as publicly verifiable evidence on misbehavior. With the exception of the DC-net protocol, all other solutions require verifying the ZKPs as well. As shown in Table 2, an AV-net has the fewest zero-knowledge proofs per participant: a constant two (i.e., one for each round). Hence, under the same evaluation conditions, the verification cost in an AV-net is the smallest among the related techniques. In the following, we will compare an AV-net with each of the past solutions in detail.

Let us first compare an AV-net with a DC-net. The DC-net protocol could be implemented with different topological designs. A fully-connected DC-net is "unconditionally secure", but suffers from the scalability problem when applied to a large system. For this reason, Chaum suggests a ring-based DC-net in [5], which presents a trade-off between security and system complexity. Recently, Wright, Adler, Levine and Shield showed that the ring-based DC-net described by Chaum (also by Schneier [17]) is easily attacked [19]. They compared different topologies of a DC-net and concluded that the fully-connected one is most resilient to attacks [19]. Hence, we compare an AV-net with the most secure form of a DC-net, i.e., a fully-connected DC-net.

As explained earlier, one of the most problematic parts in a DC-net is its key setup, which produces $O(n^2)$ keys. In the original description of the DC-net protocol, shared keys are established by *secretly* tossing coins behind menus. However, this requires multiple rounds of interaction between pairs of participants. It is slow and tedious, especially when there are many people involved. Other means to establish keys, as suggested by Chaum, include using optical disks or a pseudo-random sequence generator based on short keys [5]. However, such methods are acknowledged by Chaum as being either expensive or not very secure [5].

Our protocol replaces the key-setup phase in a DC-net with a simple one-round broadcast. This is achieved via public key cryptography. Although a DC-net can adopt a similar technique – the Diffie-Hellman key exchange protocol – to distribute keys, its use of the underlying technology is quite different from ours. Suppose a DC-net uses Diffie-Hellman to establish keys[2]. Each participant must perform $O(n)$ exponentiations in order to compute the shared keys with the remaining $n-1$ participants. (A DC-net has the desirable feature that once pairwise keys are established, the protocol can be run continuously [5]. But it might be a question how useful that feature is in practice given the collision and jamming problems.) However, our protocol requires only one exponentiation for each of the two rounds, no matter how many participants in the network (the cost of multiplication is negligible as compared to that of exponentiation).

[2] Note that in this case, a DC-net is no longer unconditionally secure, as the Diffie-Hellman key exchange essentially rests on the Decision Diffie-Hellman assumption [2].

Secure circuit evaluation is an important technique for secure Multi-Party Computation (MPC) applications. It evaluates a given function f on the private inputs $x_1, \ldots, x_n$ from n participants. In other words, it computes $y = f(x_1, \ldots, x_n)$, while maintaining the privacy of individual inputs. At first glance, it appears trivial to apply this technique to build a veto-protocol – one only needs to define f as the boolean-OR function. However, this general technique proves to be unnecessarily complex and expensive in solving specific functions [4].

Yao [20] first proposed a general solution for the secure circuit evaluation for the two-party case. Later, Goldreich, Micali, and Wigderson extended Yao's protocol for the multiparty case, and demonstrated that any polynomial-time function can be evaluated securely in polynomial time provided the majority of the players are honest [10]. This conclusion is drawn based on the general assumption of the existence of a trap-door permutation function. Although the general solution proposed in [10] uses an unbounded number of rounds, it was later shown that such an evaluation can be done using only a constant number of rounds of interaction [3]. Recently, Gennaro, Ishai, Kushilevitz, and Rabin showed that three rounds are sufficient for arbitrary secure computation tasks [14].

Although the general GMW solution is versatile, it suffers from the way this technique is evolved – by extending the general solution in the two party case to pairs in the multiparty case. This leads to the $O(n^2)$ system complexity. First, it requires pairwise private channels among participants [10], which could prove problematic especially when there are many participants. Second, it requires a large amount of traffic. Although the protocol could be completed with only three rounds [14], note that each round includes not only the broadcast of public messages, but also the transmission of private messages to everyone else through the pairwise secret channels [14]. The total amount of sent data is $O(n^2)$. Third, it is no longer resistant to collusion when more than half of the participants are compromised. In such a case, the colluders can easily breach the privacy of other inputs.

Our work shows the benefits of designing a protocol directly in the multiparty context. It has the linear complexity, requires no pairwise secret channels, and provides full protection against collusion instead of half. How to apply the underlying design principle in the AV-net protocol to compute more general functions is worth exploring in future research.

All the other techniques in Table 2 are based on the Decision Diffie-Hellman assumption [15, 11, 4]. The first rounds of those protocols are the same as in an AV-net: broadcasting ephemeral public keys. This allows more direct comparisons with an AV-net, as shown below.

Kiayias and Yung investigated the Distributed Decision Making problem, and proposed a 3-round veto protocol [15]. They used a third party – a bulletin board server – to administer the process. The bulletin board server is a common way to realize a reliable broadcast channel. However, the server is needed for some other reasons. In the Kiayias-Yung protocol, each participant publishes $O(n)$ data. The final result on the veto decision is computed from $O(n^2)$ data. In

large networks, it would be too demanding for individuals to store and compute such data. The server is a natural choice to perform the intermediary processing.

Groth modified the Kiayias-Yung veto protocol in order to reduce the system complexity [11]. His approach is to trade off round-efficiency for less traffic and computation. As a result, Groth's veto protocol allows each participant to publish a smaller amount of data, but requires participants to send their messages one after another, as one's computation depends on the result sent by the previous participant. Hence, instead of finishing the protocol in 3 rounds as in [15], Groth's veto protocol requires $n + 1$ rounds, where n is the number of participants.

Brandt studied the use of ElGamal encryption techniques for multiparty computation applications, and gave a 4-round veto protocol [4]. The performance of his solution, among others, is the closest match to ours. Its main disadvantage, however, is that it requires four rounds while ours only needs two. The difference in rounds lies in the way the veto messages are encrypted.

In Brandt's veto protocol, the first round is the same as in an AV-net: all participants broadcast ephemeral public keys. It requires one exponentiation to compute a public key. In the second round, each participant applies the standard ElGamal encryption algorithm to encrypt an explicit message: "veto" or "non-veto". Such an encryption requires two exponentiations. The third and fourth rounds are arranged to decrypt the messages, while preserving the privacy of individual inputs. It requires two and one exponentiations in each round respectively. In addition, each round requires a zero-knowledge proof per participant, which amounts to four in total. Without taking the knowledge proofs into consideration, each participant needs to performs six exponentiations in Brandt's protocol.

The novelty of our protocol is that the veto message is encrypted in a very implicit way (i.e., by raising a base to one of two different powers). As a result, the veto decision can be immediately decoded after the second broadcast. It requires only two exponentiations in total, as compared to six in Brandt's protocol. Besides computational load, the traffic generated is also far less in our protocol, due to fewer rounds.

5 Conclusion

In this paper, we propose Anonymous Veto Network (or AV-net) to allow a group of participants to compute a boolean-OR function securely. Our technique is not only provably secure, but also exceptionally efficient. Compared with past work, our solution does not require any private channels or trusted third parties; it has no message collisions, hence requires no retransmissions; being semantically secure, it provides the strongest protection of vetoer's anonymity until all the other participants are compromised; it resists robustly against jamming, hence ensures each participant's veto power; the execution of the protocol requires only two rounds, fewer than any other solutions; and finally, the computational load, the bandwidth usage, and the cost of verifying zero-knowledge proofs are also the least among the related techniques.

Acknowledgments

We thank Markus Kuhn, Ross Anderson, George Danezis, Mike Bond, Saar Drimer, Tyler Moore and other members in the Security Group, Computer laboratory, University of Cambridge, for providing helpful comments and feedbacks. Special thanks go to Lihong Yang, Nick Wolfgang and Rachel Wolfgang for helping improve the readability of this paper.

References

1. Hao, F., Zieliński, P.: A 2-round anonymous veto protocol. In: Proceedings of the 14th International Workshop on Security Protocols, Cambridge, UK (2006)
2. Boneh, D.: The Decision Diffie-Hellman Problem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 48–63. Springer, Heidelberg (1998)
3. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols. In: Proceedings of the twenty-second annual ACM Symposium on Theory of Computing, pp. 503–513 (1990)
4. Brandt, F.: Efficient cryptographic protocol design based on distributed El Gamal encryption. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 32–47. Springer, Heidelberg (2006)
5. Chaum, D.: The dining cryptographers problem: unconditional sender and recipient untraceability. Journal of Cryptology 1(1), 65–67 (1988)
6. Chaum, D.: Untraceable electronic email, return addresses, and digital pseudonyms. Communications of the ACM 24(2), 84–88 (1981)
7. Chaum, D., Evertse, J.H., van de Graaf, J., Peralta, R.: Demonstrating possession of a discrete logarithm without revealing it. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 200–212. Springer, Heidelberg (1987)
8. Chaum, D., Evertse, J.H., van de Graaf, J.: An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 127–141. Springer, Heidelberg (1988)
9. Camenisch, J., Stadler, M.: Proof systems for general statements about discrete logarithms. Technical report TR 260, Department of Computer Science, ETH Zürich (March 1997)
10. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Proceedings of the nineteenth annual ACM Conference on Theory of Computing, pp. 218–229 (1987)
11. Groth, J.: Efficient maximal privacy in boardroom voting and anonymous broadcast. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 90–104. Springer, Heidelberg (2004)
12. Golle, P., Juels, A.: Dining Cryptographers Revisited. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 456–473. Springer, Heidelberg (2004)
13. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28, 270–299 (1984)
14. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: On 2-round secure multiparty computation. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, p. 178. Springer, Heidelberg (2002)

15. Kiayias, A., Yung, M.: Non-interactive zero-sharing with applications to private distributed decision making. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 303–320. Springer, Heidelberg (2003)
16. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology 4(3), 161–174 (1991)
17. Schneier, B.: Applied Cryptography. J. Wiley and Sons, Chichester (1996)
18. Waidner, M., Pfitzmann, B.: The Dining Cryptographers in the Disco: Unconditional Sender and Recipient Untraceability with Computationally Secure Serviceability. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 690–690. Springer, Heidelberg (1990)
19. Wright, M., Adler, M., Levine, B.N., Shields, C.: The predecessor attack: an analysis of a threat to anonymous communications systems. ACM Transactions on Information and Systems Security (TISSEC) 7(4) (2004)
20. Yao, A.: How to generate and exchange secrets. In: Proceedings of the twenty-seventh annual IEEE Symposium on Foundations of Computer Science, pp. 162–167 (1986)

A Lower Bound for Round Efficiency

Theorem 5. *Without shared symmetric or asymmetric secrets between participants, any anonymous veto protocol relying on authenticated broadcast only must require at least two rounds.*

Proof. To obtain a contradiction, we assume a one-round anonymous veto protocol. Each participant holds a secret vote $v_i \in \{0, 1\}$, and has no shared secrets with others. In one round, every participant P_i broadcasts $f_i(v_i)$, where f_i is a publicly known function.

Note that the function definition f_i cannot be secret (known to P_i only). Otherwise, the value of $f_i(v_i)$ would contain no useful information to the rest of participants, and could be equivalently replaced by an arbitary value. This contradicts the veto power that P_i has on the decision making. So f_i must be a publicly known function.

The protocol allows participants to determine the Boolean-OR of all votes. Suppose every participant P_i can do so by applying a function g_i to all data available: $g_i(f_i(v_i), \ldots, f_n(v_n)) = v_1 \vee \ldots \vee v_n$. Thus participant P_i can trivially reveal the vote of another participant, say P_k, through simulating other participant's inputs as 0: $g_i(f_i(0), \ldots, f_k(v_k), \ldots f_n(0)) = 0 \vee \ldots \vee v_k \vee \ldots \vee 0 = v_k$. This contradicts the secrecy of the vote v_k, which shows that any such anonymous veto protocol requires at least two rounds.