A 2-Round Anonymous Veto Protocol

Feng Hao, Piotr Zieliński

Computer Laboratory, University of Cambridge, UK
{feng.hao, piotr.zielinski}@cl.cam.ac.uk

Abstract. The dining cryptographers network (or DC-net) is a seminal technique devised by Chaum to solve the dining cryptographers problem — namely, how to send a boolean-OR bit anonymously from a group of participants. In this paper, we investigate the weaknesses of DC-nets, study alternative methods and propose a new way to tackle this problem. Our protocol, Anonymous Veto Network (or AV-net), overcomes all the major limitations of DC-nets, including the complex key setup, message collisions and susceptibility to disruptions. While DC-nets are unconditionally secure, AV-nets are computationally secure under the Decision Diffie-Hellman (DDH) assumption. An AV-net is more efficient than other techniques based on the same public-key primitives. It requires only two rounds of broadcast and the least computational load and bandwidth usage per participant. Furthermore, it provides the strongest protection against collusion — only full collusion can breach the anonymity of message senders.

1 Introduction

Chaum introduced the *dining cryptographers problem* in 1988: three cryptographers want to find out whether NSA or one of them pays for the dinner, while respecting each other's right to make a payment anonymously [1].

In the same paper, Chaum provided a well-known solution: the dining cryptographers network (or DC-net). A DC-net uses "unconditional secrecy channels" to setup pairwise shared keys and an authenticated broadcast channel to send anonymous messages whose senders are untraceable. Details of DC-nets can be found in [1].

Despite their importance in anonymity research, DC-nets are not widely deployed for practical applications. The major problem is their requirement of *pairwise* shared keys. Setting up these keys relies on unconditionally secret channels [1]. The number of such channels grows squarely with the increasing network size, as does the total number of the shared keys. Message collisions are also problematic in DC-nets. If a collision occurs, a retransmission needs to be arranged. However, as we explain in Section 2.3, there are circumstances where retransmissions cannot resolve the collision problem. Finally, DC-nets are subject to various forms of disruptions [2]. Solutions to prevent disruptions make the system more complex.

We would like to highlight that the DC-net is not the only solution to the dining cryptographers problem. Essentially, DC-nets are designed to determine

the boolean-OR of bits contributed by participants, while preserving the privacy of individual inputs [1]. Alternatively, the circuit evaluation technique can be applied to compute the boolean-OR function securely [11, 18]. However, due to its generality, the circuit evaluation technique is expensive and impractical [10]. This will be explained in Section 3 in more detail.

We consider the dining cryptographers problem from a different perspective — suppose the three cryptographers vote against the statement: "no cryptographer has paid". If anyone vetoes, it means "one of the cryptographers has paid". Otherwise, it implies "NSA has paid". Thus, an anonymous veto protocol can solve this problem well. Several such protocol designs exist [10, 12, 13].

In this paper, we propose a new veto protocol: anonymous veto network (or AV-net). Our solution is simple and very efficient. As opposed to DC-nets, AV-nets require no secrecy channels, have no message collisions, and are more resistant to disruptions. Compared to other veto protocols [10, 12, 13], AV-nets are more efficient in nearly every aspect, such as the number of rounds, computational load and bandwidth usage. In the rest of the paper, Section 2 explains the protocol and analyzes its security properties. Section 3 examines the efficiency of the protocol and compares with prior art.

2 Protocol

Our protocol does not require any private channels or third parties. It only assumes an *authenticated broadcast channel* available to every participant. In fact, this assumption is made in all past work in this line of research [1, 10–13] (see Section 3). It suffices to know that such a broadcast channel can be realized using physical means or digital signatures [1].

The protocol setting resembles the real-life situation quite closely — when people engage in public discussion, every word uttered can be traced to its originators. How can a participant with the veto right to say "no" *anonymously* in such an open environment? Our solution is provided below.

2.1 Two-round broadcast

Let G denote a finite cyclic group of prime order q in which the Decision Diffie-Hellman (DDH) problem is intractable [3]. Let g be the generator. 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 broadcasts g^{x_i} and a knowledge proof for x_i.*

When this round finishes, each participant computes

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

	x_1	x_2	x_3	x_4	x_5
x_1		−	−	−	−
x_2	+		−	−	−
x_3	+	+		−	−
x_4	+	+	+		−
x_5	+	+	+	+	

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

Round 2 *Every participant broadcasts 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 *For the x_i and y_i defined in AV-nets, $\sum_i x_i y_i = 0$.*

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.

In the protocol, senders must demonstrate their knowledge of the discrete logarithms, namely the secrets x_i and c_i in each round respectively, without revealing them. This can be realized by using a zero-knowledge proof, a well-established primitive in cryptography [8]. Zero-knowledge proofs are commonly used in the related work in order to prevent certain attacks [10–13]. Several zero-knowledge proof techniques have been presented in past literature [5–8]. One can use, for example, Schnorr's signature [7], which is suggested in Brandt's veto protocol [10]. Schnorr's signature is a suitable choice because it is short, non-interactive, and reveals nothing except the one bit information about the truth of the statement: "the sender knows the discrete logarithm" [7].

For example, let H be a *publicly known* 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. One should note that here the participant index i is unique and known to all. Adding i inside the hash function can effectively prevent the replay of this signature by other participants. More details on Schnorr's signature and other zero-knowledge proof techniques can be found in [7, 8].

There is a variant of our protocol, in which there is no need to use any zero-knowledge proofs. Instead, participants need to commit to their announcements before each broadcast round. This can be easily realized in the physical world — for example, all people write down their numbers on the paper before the broadcast round. However, in computer networks, this often requires additional rounds to send the results of applying a one-way hash function. It can prove costly if network communication is expensive.

2.2 Semantic security

In order to analyze the security of our technique, we now examine the protocol more closely: in the first round, all participants announce their public keys g^{x_i}; in the second round, each uses a collaborative form of everyone else's public key to encrypt a one-bit message and announces the ciphertext.

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* [4]. This is equivalent to showing that under the hard-problem assumption, ciphertext is indistinguishable to observers [4]. First, we need to evaluate the resistance of our protocol against collusion.

Definition 2 *In a collusion attack, a subset of the participants are compromised, with their secrets x_i revealed.*

The *full collusion* against P_i involves all other participants in the network. Any anonymous veto protocol, by nature, cannot preserve the vetoer's anonymity under this circumstance. However, in practice, it is impossible to have all participants — who are mutually mistrustful — colluding against just one in an anonymous network; there would be no point for that person to stay in the network [1]. Hence, a more realistic attack is the *partial collusion*, which involves only some of the participants.

In AV-nets, 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 3 *In AV-nets, y_i is a secret of 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 4 *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. The secret x_i is chosen randomly by P_i. Lemma 3 shows that y_i is a random value, unknown to attackers. DDH states that one cannot distinguish between $g^{x_i y_i}$ and a random value in the group such as $g^{r_i y_i}$ [3].

The above theorem states that attackers cannot break the anonymity of the individual participant without full collusion. The one-bit message on the veto decision is decoded from the multiplication of all ciphertexts. The question is, whether additional information could be decoded as well. In our protocol, since the vetoer knows his random input, it is possible that he could derive the extra information: whether or not he is the only one who vetoed. If this is of much concern, there are solutions proposed in [13]. However, the derived information is only *one bit* and tells nothing about who else vetoed, nor how many vetoers there are. For this reason, this issue is generally not considered in the related work [1, 12, 13] — for example, a collision in the DC-net "leaks" the information that an even number of participants are sending messages, but that is not seen as a threat [1].

2.3 Attacks

Collusion is a common attack against anonymity [1,10–13]. Our protocol provides the strongest protection against such an attack — only full collusion can breach the anonymity of message senders.

Another attack makes use of message collisions. In the broadcast round of a DC-net, each participant sends one bit: b_i. The anonymous message received by everyone is the XOR of all the sent bits [1]. A known weakness in DC-nets is that an even number of messages would cancel each other out, forcing retransmissions [1]. Collisions not only reduce the transmission efficiency, but also can be exploited by attackers to jam the sent messages. For example, the last participant (an attacker) can announce $\sum_{i=1}^{n-1} b_i \mod 2$. Then, the overall message will always be '0'. A retransmission cannot resolve this problem as long as the attacker is the last announcer.

This collision attack may be viewed as one instance of disruption. Chaum suggested a few countermeasures to prevent disruptions [1]. Those relevant to this particular attack are twofold: broadcasting simultaneously on different frequencies or committing to output before broadcast [1]. However, both methods require additional rounds, which would significantly reduce the protocol efficiency.

related work	pub year	round no	broad-cast	pvt ch	colli-sion	3rd pty	collu-sion	security reliance	total traffic	total comp
GMW [11]	1987	$O(1)$	yes	yes	no	no	half	trapdoor	$O(n^2)$	$O(n^2)$
Chaum [1]	1988	2+	yes	yes	yes	no	full	uncond	$O(n^2)$	$O(n^2)$
KY [12]	2003	3	yes	no	no	yes	full	DDH	$O(n^2)$	$O(n^2)$
Groth [13]	2004	$n+1$	yes	no	no	yes	full	DDH	$O(n)$	$O(n)$
Brandt [10]	2005	4	yes	no	no	no	full	DDH	$O(n)$	$O(n)$
AV-net	—	**2**	**yes**	**no**	**no**	**no**	**full**	**DDH**	$\boldsymbol{O(n)}$	$\boldsymbol{O(n)}$

Table 2. Comparison to the past work

In contrast, our protocol is resistant to collisions, whether intentional or not. First consider the situation where more than one participant sends the "veto" message. Each one randomly chooses r over $\mathbb{Z}_q$. Given a cyclic group with big q (e.g., 1024-bit), the likelihood of message collisions, which results in $\prod_i g^{c_i y_i} = 1$, is negligible. In addition, intentional collisions are prevented by our protocol. Let $z = \prod_{i=1}^{n-1} g^{c_i y_i}$. The last announcer cannot send $1/z$ to jam the veto message — to provide the required knowledge proof for c_i, he would have to solve the Discrete Logarithm problem $(g^{y_i})^{c_i} = 1/z$, which is believed to be intractable [3].

3 Performance

There are related techniques proposed in past literature. Table 2 presents a comparison between our protocol and the previously proposed solutions.

Let us first compare AV-nets with DC-nets. Both protocols determine the boolean-OR of bits from a group of participants in such a way that message senders are untraceable. DC-nets could be implemented with different topological designs. A fully-connected DC-net is unconditionally secure. But it suffers from the scalability problem when applied to a large system. For this reason, Chaum suggests a ring-based DC-net in [1], 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 [14]. They compared different topologies of DC-nets and concluded that the fully-connected DC-net is most resilient to attacks [14]. Hence we compare AV-net only with the most secure form of DC-net, i.e., the fully-connected one.

A DC-net has two phases of operation: key setup and a one-round broadcast. The key setup phase — which produces $O(n^2)$ keys — is usually the problematic part in practice. In the original description of a DC-net, shared keys are established by *secretly* tossing coins behind menus. However it requires multiple rounds of interaction between pairs of participants. It is very 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 [1]. However, such methods are acknowledged by Chaum as being either expensive or not very secure [1].

Our protocol replaces the problematic 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[1]. Each participant must perform $O(n)$ exponentiations in order to compute the shared keys with the remaining $n-1$ participants. However, our protocol requires only one exponentiation for each of the two rounds. The computational load for each participant remains unchanged even when applied to a larger system (the cost of multiplication is negligible as compared to that of exponentiation).

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 for solving a specific function like the Boolean-OR [10].

Yao [18] 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 [11]. This conclusion is drawn based on the general assumption of the existence of a trap-door permutation function. Although the general solution proposed in [11] 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 [15]. Recently, Gennaro, Ishai, Kushilevitz, and Rabin showed that three rounds are sufficient for arbitrary secure computation tasks [16].

Although the GMW solution to the circuit evaluation problem is more versatile than ours, it is much less efficient when used in a veto protocol. First, the GMW protocol requires pairwise private channels among participants [11], which has the complexity of $O(n^2)$. Second, 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. Third, it requires a large amount of traffic. Although the protocol could be completed with only three rounds [16], 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 secrecy channels [16]. The total amount of sent data is $O(n^2)$.

Kiayias and Yung investigated the Distributed Decision Making problem, and proposed a 3-round veto protocol [12]. 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

[1] 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 [3].

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 [13]. 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 [12], 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 [10]. 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 AV-nets: all participants broadcast their 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. Without taking the knowledge proofs into consideration, each participant needs to performs six exponentiations in total.

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.

4 Conclusion

In this paper, we propose the Anonymous Veto Network (or AV-net) to solve the dining cryptographers problem. Several solutions in past work are reviewed, ranging from DC-nets and circuit evaluation techniques proposed nearly twenty years ago, to several anonymous ("private") veto protocols published in recent years. We show that our solution achieves semantic security and that the anonymity of message senders is preserved unless all other participants are compromised. In comparison with other methods, AV-net is more efficient in many aspects. It does not require any private channels or third parties; it has no message collisions, hence requires no retransmissions; it needs only two rounds of broadcast, fewer than any other solution; and the required computational load

and bandwidth usage per participant are the least among the related work. Furthermore, there is very little room for improvement in each of these aspects. Its efficiency is close to the best we can possibly achieve under the security assumption of Decision Diffie-Hellman (DDH).

Acknowledgments

We thank Markus Kuhn for constructive discussions. We thank Ross Anderson, George Danezis, Mike Bond, Saar Drimer, Tyler Moore and other members in the Security Group 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. D. Chaum, "The dining cryptographers problem: unconditional sender and recipient untraceability," *Journal of Cryptology*, Vol. 1, No. 1, pp. 65–67, 1988.
2. P. Golle and A. Juels, "Dining Cryptographers Revisited," Eurocrypt'04, LNCS 3027, pp. 456–473, 2004.
3. D. Boneh, "The decision Diffie-Hellman problem," *Proceedings of the Third International Symposium on Algorithmic Number Theory*, LNCS 1423, pp. 48–63, 1998.
4. S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, Vol. 28 pp. 270–299, 1984.
5. D. Chaum, J.H. Evertse, J.V.D. Graaf and R. Peralta, "Demonstrating possession of a discrete log without revealing it," Advances in Cryptology Crypto'86, LNCS 263, pp. 200–212, 1987.
6. D. Chaum, J.H. Evertse and J.V.D Graaf, "An improved protocol for demonstrating possession of a discrete logarithm and some generalizations," Advances in Cryptology Eurocrypt'87, LNCS 304, pp. 127–141, 1988.
7. C.P. Schnorr, "Efficient signature generation by smart cards," Journal of Cryptology, Vol. 4, No. 3, pp. 161–174, 1991.
8. J. Camenisch and M. Stadler, "Proof systems for general statements about discrete logarithms," Technical report TR 260, Department of Computer Science, ETH Zürich, March 1997.
9. A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," Proceedings on Advances in Cryptology, Crypto'86, LNCS 0263, pp. 186–194, 1987.
10. F. Brandt, "Efficient cryptographic protocol design based on distributed El Gamal encryption," Proceedings of the 8th International Conference on Information Security and Cryptology (ICISC), to appear in LNCS, 2005. The draft is available from `http://www7.in.tum.de/~brandtf/studies.shtml`
11. O. Goldreich, S. Micali and A. Wigderson, "How to play any mental game or a completeness theorem for protocols with honest majority," Proceedings of the nineteenth annual ACM Conference on Theory of Computing, pp. 218–229, 1987.
12. A. Kiayias and M. Yung, "Non-interactive zero-sharing with applications to private distributed decision making," Financial Cryptography 2003, LNCS 2742, pp. 303–320, 2003.

13. J. Groth, "Efficient maximal privacy in boardroom voting and anonymous broadcast," Financial Cryptography 2004, LNCS 3110, pp. 90–104, 2004.
14. M. Wright, M. Adler, B.N. Levine, and C. Shields, "The predecessor attack: an analysis of a threat to anonymous communications systems," ACM Transactions on Information and Systems Security (TISSEC), Vol. 7, No. 4, 2004.
15. D. Beaver, S. Micali and P. Rogaway, "The round complexity of secure protocols," Proceedings of the twenty-second annual ACM Symposium on Theory of Computing, pp. 503–513, 1990.
16. R. Gennaro, Y. Ishai, E. Kushilevitz and T. Rabin. "On 2-round secure multiparty computation," Crypto 2002, LNCS 2442, pp. 178–193, 2002.
17. B. Schneier, *Applied Cryptography*, J. Wiley and Sons, 1996.
18. A. Yao, "How to generate and exchange secrets," Proceedings of the twenty-seventh annual IEEE Symposium on Foundations of Computer Science, pp. 162–167, 1986.