CliqueNet: A Self-Organizing, Scalable, Peer-to-Peer Anonymous Communication Substrate

Emin Gün Sirer Milo Polte Mark Robson

{egs, mpolte, robson}@cs.cornell.edu

Abstract

Anonymity is critical for many networked applications. Yet current Internet protocols provide no support for masking the identity of communication endpoints. This paper outlines a design for a peer-to-peer, scalable, tamper-resilient communication protocol that provides strong anonymity and privacy. Called CliqueNet, our protocol provides an information-theoretic guarantee: an omnipotent adversary that can wiretap at any location in the network cannot determine the sender of a packet beyond a clique, that is, a set of k hosts, where k is an anonymizing factor chosen by the participants. CliqueNet is resilient to jamming by malicious hosts and can scale with the number of participants. This paper motivates the need for an anonymous communication layer and describes the self-organizing, novel divide-and-conquer approach that enables CliqueNet to scale while offering a strong anonymity guarantee. CliqueNet is widely applicable as a communication substrate for peer-to-peer applications that require anonymity, privacy and anti-censorship guarantees.

1 Introduction

Strong anonymity and privacy guarantees are critical for many offline, real-world applications. Whether casting a ballot in a voting booth, engaging in a cash-based financial transaction, or getting tested for certain medical conditions, people expect that the transaction by itself will not reveal their identity. Such transactions require *strong anonymity,* where a party to a transaction needs to unconditionally hide their identity from other participants and possible eavesdroppers. Many other applications require *strong privacy*, where participants release their identity to other parties of their choice, but need to cloak it from unauthorized interceptors. For instance, whistleblowers, witnesses, patients, press sources, attorneys and clients, among many others, call for protection from third parties who might deduce their identity via traffic analysis. Indeed, anonymity and privacy are deemed so indispensable for a functional society that many countries have codified special protections into law. In addition, the real world provides many viable channels, ranging from post office boxes to anonymizing brokers, for carrying out private transactions without betraying the participants' identity.

Despite the critical role anonymous and private communication plays in the offline world, the state of user privacy in the online world is grim. Current Internet networking protocols provide no support for masking the identity of communication endpoints. An adversary with access to router nodes can monitor traffic patterns and harvest IP addresses. Tracking software, such as Carnivore/Echelon [FBI01,FAS01,RG01], can be used to map IP addresses back to individual users. While encryption schemes, like SSL [FKK95], can make it computationally difficult for attackers to decipher *what* was sent, they cannot hide *who* sent it. The situation is particularly problematic when governments and corporations engage in online monitoring and censorship, as the current set of digital communication protocols enable user tracking at an unprecedented scale.

In this paper, we describe the design of a peer-to-peer, self-organizing, scalable anonymous communication protocol, called CliqueNet. CliqueNet has three critical properties: (1) it unconditionally hides the identity of the source and destination of a packet, even from attackers with arbitrary wiretapping capabilities, (2) it scales well with increasing numbers of hosts, and (3) it is resilient against malicious and disruptive participants. The central abstraction provided by CliqueNet is that of an anonymous communication channel that supports a completely anonymous broadcast operation, as well as a sender-anonymous, efficient unicast primitive. This anonymous dial tone is akin to an Ethernet carrier, and supports traditional internetworking protocols such as TCP. In short, CliqueNet is a practical, scalable, and robust protocol, which can serve as a modular communication substrate for peer-to-peer applications that require strong anonymity and privacy guarantees.

CliqueNet differs from previous anonymous communication protocols in several important ways. Previous implementations of anonymous communication protocols and identity protection schemes have relied principally on source rewriting [C81, RR98, RSG98, GW97, CSW+00, G01] – in essence, every router in a chain replaces the source address in a packet with its own, thereby obfuscating the originator's identity. While such schemes are simple and scalable, they cannot provide strong anonymity guarantees. A powerful adversary that can capture traffic within and around the anonymizing network can perform statistical traffic analyses and corroborate the identity of users with sent packets. Further, such schemes are fragile, as a node that stops transmitting for any reason disrupts the communication path for all traffic that was routed through it. A proposed, heavy-duty alternative for anonymous communication, called DC-nets [C88], addresses the traffic analysis problem. DC-nets provide an information-theoretic guarantee that even an observer that captures every packet cannot determine the originator. However, DC-nets do not scale: the aggregate bandwidth of a DC-net follows $O(1/N^2)$ (Figure 1). Consequently, no practical implementation of DC-nets has been reported in the literature to date.

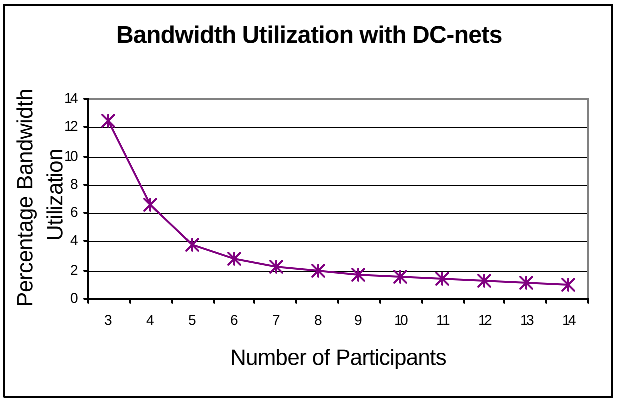

Figure 1. Traditional DC-nets scale poorly with increasing numbers of participants.

CliqueNet combines the strong anonymity and privacy properties of DC-nets with a divide-and-conquer approach that enables the system to scale well with increasing numbers of hosts. CliqueNet automatically constructs small anonymizing cliques that use an extended version of the basic, strong DC-net mechanism. CliqueNet provides an ad hoc routing protocol for routing packets between cliques while preserving the identity of the endpoints. The protocol also embodies cryptographic commitments, join certificates, and disruption proofs to identify and exclude misbehaving hosts. The overall goals of CliqueNet, and the techniques used to achieve them, are:

- ***Strong Anonymity****:* The system should provide strong, that is, computationally insurmountable, mechanisms to guard the identities of participants. CliqueNet builds upon the information-theoretic guarantees of DC-nets - even an adversary with access to every internal packet cannot determine the identities of senders and receivers.
- ***High Scalability***: The communication protocol should achieve performance that does not degrade significantly as more participants join the network. CliqueNet automatically divides the network into smaller cells, or *cliques*, and uses ad hoc routing to route for inter-clique communication.
- ***Robustness****:* Anonymous communication protocols need to be especially robust against denial of service attacks and malicious hosts, as the anonymous medium makes it easier for disruptive hosts to launch Byzantine with impunity. CliqueNet provides irrefutable, non-forgeable proofs to identify disruptive nodes and exclude them from the network.

This work makes three contributions. First, it proposes a novel scheme for achieving strong anonymity and scalability in the same system by combining DC-nets and ad hoc routing protocols via automatic clique formation. Second, it introduces the concept of a disruption proof to allow legitimate clique members to detect and exclude misbehaving hosts, without allowing malicious hosts to forge such certificates. Finally, it presents preliminary measurements from a DC-net implementation to underscore the need for an anonymity protocol that scales well and achieves high bandwidth.

A completely anonymous, practical communication protocol touches upon many high-level issues related to identity management, involving naming, persistence of reputations, accreditation, non-repudiation, and so forth. In this paper, we focus on the low-level transport protocol and the associated techniques we use to simultaneously achieve strong anonymity, scalability and tamper-resilience.

The rest of this paper is structured as follows. In the next section, we describe the basic DC-net scheme on which CliqueNet builds. Section 3 describes the self-organizing clique formation algorithm. Section 4 presents the wire-protocol we use to encode anonymous bits. Section 5 describes how CliqueNet addresses malicious and disruptive hosts and Section 6 concludes.

2 DC-Nets

CliqueNet is based on Dining-Cryptographer networks, or DC-nets, originally suggested by Chaum in [C88]. DC-nets propagate a bit of information in the following way: suppose we have two participants, Alice and Bob, one of whom (e.g. Bob) would like to communicate a one-bit message to Charlie, but Alice and Bob want to hide the identity of the message originator. They first toss a coin in secret. Alice sends the truthful result of the coin flip to Charlie. Bob, on the other hand, reports the true result of the coin toss only if he wants to transmit a 0. If he wants to transmit a 1, Bob lies about the coin flip. Charlie deciphers the message by XOR'ing the values sent by Alice and Bob. If they both call out heads or tails, they are both telling the truth and the one-bit message is a zero; otherwise, one of them is lying, and the one-bit message is a one. Since Charlie does not know if it was Alice or Bob who lied about the coin toss, he can never determine who sent the message. This security guarantee is strong and information-theoretic: no amount of computational power can help Charlie determine that it was Bob who sent the message.

Turning this basic idea into a general scheme for communication between arbitrary numbers of hosts is straightforward. Typically, all participants will require anonymity, which can be achieved by arranging all the participants in a fully connected graph. Every pair of nodes with an edge between them share a virtual coin. The coin tosses are generated in blocks by a pseudo-random number generator. Instead of wasting bandwidth to exchange the pseudo-random stream, members use a standard secure key exchange protocol [DH76, PH78] to perform a pairwise exchange of initial seeds. Figure 3 illustrates a simple DC-net in operation where a 0 is being transmitted. The marked values on the edges represent the coin tosses. Each node transmits the XOR of the coin tosses, as indicated next to node names, to every other node. Upon receiving the reported values from B and C, A can compute $0 \otimes 1 \otimes 1 = 0$ as the transmitted bit for that round. If C wants to transmit a 1, it would simply invert the value it reports and send a 0 to A and B.

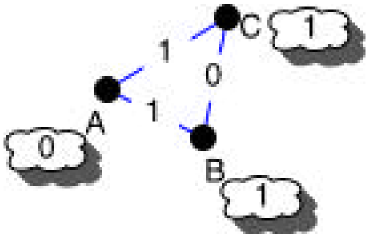

Figure 2. The basic DC-net scheme. The data bit being transmitted over the anonymous channel is a 0.

DC-net transmissions require that only a single node transmits at a given time. If an even number of parties lie about a coin toss to transmit a 1, they will end up transmitting a zero instead. Consequently, data transmission, which occurs in slots, is preceded by a slot reservation phase to ensure that only a single host is transmitting at any given time. This requirement that only a single node transmit at a given time slot forms a potential vulnerability for DC-nets. A malicious host could jam the channel by transmitting out of turn, somewhat like inducing collisions on an Ethernet. Worse, the anonymity properties of DC-nets shield the malicious attacker – none of the other nodes would be able to determine which node transmitted without a reservation in that round. Prior work [WP90] has examined this problem and proposed a solution, based on pre-commitments to data using MD5 hashes of the keystream, that involves a four-phase protocol to detect malicious hosts via traps. We adopt this approach, and extend it with irrefutable proofs and certificates to identify and exclude malicious hosts. In the next section, we describe the network topology control protocol, and later discuss the resulting wire protocol for encoding anonymous bits in detail.

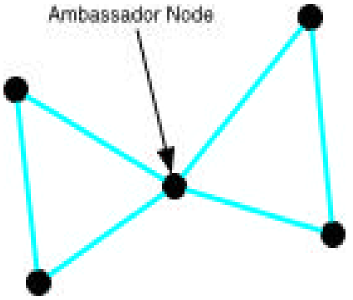

Figure 3. An ambassador node.

3 The CliqueNet Topology Control Protocol

CliqueNet achieves scalability by automatically partitioning the network into smaller DC-nets of between 3 and 5 participants each, called *cliques*. A single client may be a member of multiple cliques, joining them together. Such a client is called an *ambassador* (Figure 3). Ambassadors act as routers and are responsible for transmitting messages from clique to clique. Ambassador nodes propagate broadcast packets by copying them from the source clique to every other clique to which they belong. A cache of the MD5 contents of recently sent packets avoids duplication and routing loops, while preventing denial of service attacks. The restrictions on the packet format are minimal. A destination address identifies if the packet is to be broadcast or unicast, but the latter is purely an optimization. A source can, for instance, send broadcast packets encrypted with a destination's public key and a hash – destinations can decrypt every packet, and quietly accept those for which the hash matches. Ambassador nodes implement an ad hoc routing algorithm to route unicast packets efficiently [JM96, PR99, see RT99 for a survey]. A single destination field in the packet identifies the intended destination mnemonically; we do not rely on IP addresses for routing and are working on secure alternatives.

CliqueNet achieves low transmission and reorganization latency by ensuring that every clique is fully connected. This makes reorganization of the clique easy when a clients leaves. It also means that as long as any two members of a clique remain uncompromised, they share a secret key to protect their anonymity.

Our topology control protocol provides maximum flexibility while enforcing the size invariant on cliques. We enable nodes to join the network by simply contacting any other node that is already a part of a clique. If a clique has less than five members, the new node is simply added to that clique. When a node joins a clique with five members, we split that clique into a three-node and a four-node clique, with an overlapping ambassador, as shown in Figure 4.

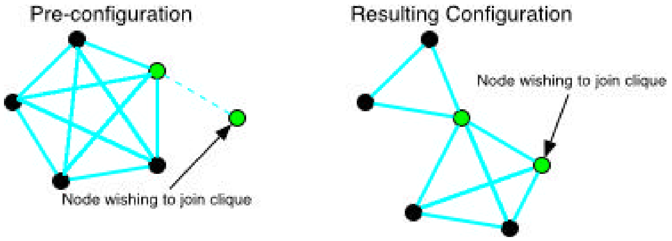

Figure 4. Clique reconfiguration with topology control.

Each clique has a unique, randomly chosen, 128-bit identifier, and a 32 bit epoch number that is incremented by one as nodes join or leave. Nodes joining a clique need to present a certificate of merit (described in Section 5). They also need to sign a join certificate indicating that they were members of the clique at the given epoch number.

When a node leaves a clique, it must be snipped out of the network. Since the cliques are fully connected, no reorganization is needed unless the clique has dropped to fewer than three members. In the case of below-par cliques, all nodes, which are not ambassadors to another clique, are responsible for rejoining the network on their own via a retry mechanism. If a node is an ambassador, it performs a secure census to detect network partitions and attempts to heal the network without disturbing its dependent cliques. When there is a potential split, the ambassadors send out a census to count the number of nodes in the network. If a node finds that it is in the minority partition, it can elect to rejoin the larger half directly by contacting a known node in the other partition.

4 The Wire Protocol

Communication in CliqueNet occurs in rounds. Each round consists of a series of phases, which ultimately end up transferring 8K blocks of data in a series of slots. The following discussion outlines the phases, and provides our implementation decisions and rationale; [C88, WP90] provide further details.

Reservation phase – In the reservation phase, each participant anonymously reserves two blocks of the latter two phases, by choosing bits i and j at random and anonymously sending a 1 in those locations. All clients must reserve two slots regardless of whether or not they intend to communicate during the present round. We use a bit vector of 4096 bits, determined by the birthday paradox to yield a collision rate of 1% for 5-member cliques. Collisions in this phase are detected by comparing the number of reserved slots to the number of participants, and simply prompt a repeat of the phase. Slots that are not claimed by anyone are eliminated from the protocol entirely and do not waste bandwidth.

Commitment phase – In this phase, clients anonymously commit to what they will be sending during the subsequent slots. They publicize the MD5 hash of the data blocks they will be using in the slots that they did not reserve. They may also mark some slots as *traps*, periods during which dummy messages are sent and collisions from other hosts will prompt an investigation without fear of exposing any valuable data. The commitment phase ensures that nodes will be caught if they transmit on the anonymous channel during someone else's reservation.

Transmit authorization vote – This phase exists to ensure that there were no collisions in the commitment phase, and needs to be performed non-anonymously. Every member of the clique sends a 1-byte message that signifies whether or not all is well with the announcement phase and therefore transmission should occur or if instead, an investigation should begin.

Transmission phase – This phase transmits data in 8K slots. Participants send a message during each of their reserved blocks, or a dummy message if there is nothing to send. Note that if an attacker attempts to disrupt the protocol, the reserved owner of that slot can request an investigation, if it has previously marked the slot as a potential trap.

Since CliqueNet requires clients to consume pseudorandom numbers to encrypt messages even when they're not sending, the entropy of a key could quickly be exhausted if clients continually transmit at full speed. CliqueNet conserves key space by slowing down the round frequency in the absence of a signal, sent over the anonymous channel at each round, which indicates that the round frequency should be increased. This is analogous to Ethernet backoff – the round frequency drops exponentially up to a threshold to conserve keys.

5 Malicious Hosts

Malicious nodes are detected through a distributed investigation. Investigations can be initiated by any host that discovers that a message it sent out in a time slot was not the message that was received, i.e. there was an observed collision. The investigation forces all nodes to reveal their keys for that slot. If, for any host, the hash of the actual transmitted values for an unreserved slot does not equal the hash of the keystream announced in the commitment phase, that host is an attacker. If a node calls for an investigation but cannot prove that it owned the slot during which the investigation was prompted, it is also engaging in an attack. In either case, all nodes independently drop that node from the clique and file a complaint with a distributed database. The complaint includes the clique id, the epoch, the clique cardinality during that epoch, the join certificates for all clique members, and the signed key certificates for the round in question. Note that malicious nodes cannot engage in denial-of-service attacks by manufacturing false complaints, as they would have to forge a join certificate. And a successful complaint will require the participation of all clique members. While it is possible for a clique of colluding nodes to manufacture false complaints, it is also possible for such cliques to decrypt the anonymous messages sent by a node. We therefore do not view this as a practical limitation – the ability to join cliques at specified locations enables nodes to avoid such situations.

Detected malicious hosts are kept out of the network via a blacklist kept in the distributed database. This database provides signed certificates of merit to nodes that wish to join the network that have not been blacklisted. Each node in CliqueNet requires a public-private key pair for secure signatures. Thus nodes are identified in the blacklist by their public keys, instead of IP addresses. This approach slows denial-of-service attacks down by necessitating that the node pick a new public key after every attack, at which time additional computational challenges can be posed to the node. While adversaries with large computing resources can still launch DOS attacks, we believe that identifying, excluding and forcing malicious hosts to spend extra resources is the maximum possible in the Internet. Note that we leave the exact implementation of the database unspecified – we currently use a centralized host, and are looking into techniques for distributing the data and signatories in a decentralized way [ZSR00,RD01,SMK+01].

6 Conclusions

We have outlined the design and implementation of a protocol that guarantees strong anonymity and privacy for its participants, protects the network from disruptive hosts, and scales well with the number of participants. The abstraction that CliqueNet provides is general, well-contained and modular. These properties make it suitable as an anonymous communication substrate in peer-to-peer applications that require strong anonymity and privacy.

References

[C81] D. Chaum. Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. Communications of the ACM, pp. 84-88, 24(2), Feb 1981.

[C88] D. Chaum. The Dining Cryptographers Problem. *Journal of Cryptology*, pp. 65-75, 1(1), 1988.

[CSW+00] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, 2000.

[DH76] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, pp 644-654, 22(6), Nov. 1976.

[JM96] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[FKK95] A. Freier, P. Karlton, and P. Kocher. The SSL protocol version 3, December 1995.

[G01] The Gnutella Protocol. http://www.gnutelladev.com/protocol/gnutella-protocol.html, Dec 2001.

[GW97] I. Goldberg and D. Wagner. TAZ Servers and the Rewebber Network: Enabling Anonymous Publishing on the World Wide Web. Unpublished manuscript, May 1997, http://www.cs.berkeley.edu/~daw/cs268/.

[FAS01] Federation of American Scientists. The Echelon Program. http://www.fas.org/irp/program/process/echelon.htm, Dec 2001.

[FBI01] FBI, Carnivore Diagnostic Tool. http://www.fbi.gov/hq/lab/carnivore/carnivore.htm, Dec 2001.

[G01] R. Graham. Carnivore Frequently Asked Questions, http://www.robertgraham.com/pubs/carnivore-faq.html, Dec 2001.

[PR99] C. E. Perkins and E. M. Royer. Ad hoc On-Demand Distance Vector Routing. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999.

[PH78] S. C. Pohlig and M. E. Hellman. An Improved Algorithm for Computing Logarithms in GF(p) and Its Cryptographic Significance. *IEEE Transactions on Information Theory*, pp 106-111, 24(1), Jan. 1978.

[RSG98] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Specific Areas in Communications*. 16(4), 1998, pp. 482-494.

[RR98] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, pp. 66-92, 1(1), 1998.

[RT99] E. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine*, April 1999, 46-55.

[RD01] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001).* Heidelberg, Germany, November 2001.

[SMK+01] I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan. Chord: A Peer-to-Peer Lookup Service for Internet Applications. ACM SIGCOMM Conf., San Diego, CA, September 2001.

[WP90] M. Waidner and B. Pfitzmann. The Dining Cryptographers in the Disco: Unconditional Sender and Recipient Untraceability with Computationally Secure Serviceability. In *Advances in Cryptology EUROCRYPT 89*, LNCS-434, Springer-Verlag, 1990.

[ZSR00] L. Zhou, F. B. Schneider, and R. van Renesse. “COCA: A Secure Distributed On-line Certification Authority”. *Technical Report 2000-1828*, Department of Computer Science, Cornell University, Ithaca, NY USA. December 2000.