PRIVATE INFORMATION RETRIEVAL AND DISSENT

ALBERT KWON, ALEX COLE, DAVID FIELD, AND DAVID LAZAR

1. Introduction

Dissent is an anonymous communication system that offers much stronger anonymity guarantees than Tor. Dissent clients efficiently publish anonymous messages, but any response must be broadcast to all clients. This approach does not scale to many clients or large responses. We've developed a private information retrieval scheme that enables clients to receieve responses without the bandwidth burden of broadcasting. Our protocol reduces bandwidth usage from quadratic in the number of clients to linear in the number of clients, while maintaining the same anonymity guarantees as Dissent.

2. Protocol

Our protocol is composed of two phases, request and retrieval. There are C clients, $\{c_i\}$, and S anonymity servers, $\{s_i\}$. The clients utilize the servers to anonymously browse the web. During the request phase, clients submit requests to servers, which then make requests to the outside web and retrieve the responses. During the retrieval phase, the clients retrieve the webpages they requested from the servers, utilizing our private information retrieval scheme. As long as at least one server is honest, the privacy of requests and retrievals is computationally protected. This means that a group of colluding clients and servers cannot identify who made a request with non-negligible advantage over randomly guessing one of the honest clients.

The request phase is performed using DC-nets and verifiable shuffle. Refer to the Dissent papers for a detailed discussion of this phase ([1], [2]). Note that the request phase does not have be initiated by any client, it is run frequently so clients can make requests without revealing their desire to make a request.

3. Private Information Retrieval

After the request phase, each server, s_i, holds a list of all W requested webpages, w_j. (Some clients may have not requested any webpage.) Websites are padded out to a fixed length. Very large websites can be chunked and transferred over multiple rounds. The following procedure is used to transfer the websites to the clients:

(1) Each client c_i will send a mask $m_{i,j}$ to each server s_j such that $\bigoplus_j m_{i,j} = m_i$ where m_i is a mask with only the index $\pi(i)$ of desired website set. The first $S-1$ masks are drawn randomly, and then the last mask is chosen to given the desired sum.

(2) Each server s_j computes the response $r_{i,j}$ to mask $m_{i,j}$ by computing the xor sum of the websites that are 1's in the mask,

$$r_{i,j} = \bigoplus_k m_{i,j}[k]w_k$$

(3) Each client c_i computes the xor of the responses it receives, $\bigoplus_j r_{i,j}$. This will be the desired website, since

$$\begin{aligned}\bigoplus_j r_{i,j} &= \bigoplus_j \bigoplus_k m_{i,j}[k]w_k \\ &= \bigoplus_k \bigoplus_j m_{i,j}[k]w_k \\ &= \bigoplus_k m_i[k]w_k \\ &= w_{\pi(i)}\end{aligned}$$

3.1. **Example.** For example, if there are three websites and three servers, and client 1 wants the second website, it could make the following requests:

$$\begin{aligned}m_{1,1} &= 011 \\ m_{1,2} &= 101 \\ m_{1,3} &= 100 \\ m_1 &= m_1 \oplus m_2 \oplus m_3 = 010\end{aligned}$$

The servers would respond with

$$\begin{aligned}r_{1,1} &= w_2 \oplus w_3 \\ r_{1,2} &= w_1 \oplus w_3 \\ r_{1,3} &= w_1\end{aligned}$$

Then the client computes the xor of all the server responses, which is w_2, as desired.

3.2. **Anonymity.** Let us now prove that our PIR scheme preserves information theoretic anonymity as long as at least one server is honest. Note that this is a strictly stronger guarantee than is offered by the DC-net request step, which only offers computational anonymity.

Theorem 1. *Our PIR protocol achieves information theoretic anonymity against colluding attackers, as long as at least one server is honest.*

Proof. Suppose some $S-1$ servers are colluding. This means they have access to the $S-1$ masks they receive.

Recall that we generate the masks for client c_i by drawing the first $S-1$ masks randomly and then choosing the final mask such that $\bigoplus_j m_{i,j} = m_i$. The procedure draws samples uniformly from the set of S-tuples of masks that xor to m_i. We could equivalently draw random masks for the $S-1$ colluding servers, and then choose the mask of the honest server such $\bigoplus_j m_{i,j} = m_i$, and this would generate the same distribution over S-tuples of masks. So, the malicious servers are receiving $S-1$ random masks. Clearly, this can give them no information about m_i. Hence our protocol achieves information theoretic anonymity.

□

3.3. **Bandwidth usage.** If each website is B bits large, then this scheme results in $SC(B+W)$ bits of communication. In practice, $B >> W$, since websites are greater than 10KB and the total number of requests per round is generally less than 10^5. As such, we can simplify to SCB bits of communication. In contrast, broadcasting all websites to all clients uses CWB bits of communication. The number of websites requested is linear in the number of clients, so this reduces communication from quadratic in the number clients to linear in the number of clients. (Actually, there is still a quadratic term, but the constant factor in front of it is reduced by a factor of $B/S > 10000$.) Some typical numbers are $S = 10$, $C = 1000$, $W = 100$, (10% of clients make a request), and $B = 100000$. With these numbers, our protocol uses 10^9 bits, whereas the original protocol uses 10^{10} bits.

Our PIR scheme outperforms the original protocol when $W > S$. When, $W \leq S$, the servers should fall back on the original broadcast technique. Our scheme is best suited for environments where clients are making many requests. For example, if some clients are streaming or downloading large amounts of data, they will each be continuously requesting. As such, W will be high, and the PIR scheme will dramatically decrease bandwidth usage.

Importantly, this protocol can handle many clients. Large client pools give greater anonymity and protection against intersection attacks. For example, with the previous numbers, if each server has a 10Gbps link, and rounds are run every seconds, a total of 10^5 clients can participate in the DC-net, whereas only $3 \cdot 10^3$ could participate with the old protocol.

Our protocol also makes it possible to increase bandwidth at the cost of latency. By increasing the delay between rounds, the number of websites W is increased, which increases the efficiency the protocol. By doubling W, the number of websites transferred is doubled, with a negligible increase to the number of bits of communication. This allows graceful handling of heavy loads.

4. Implementation

To show our protocol in action, we built a system which utilizes DC-nets and our new PIR scheme to enable anonymous web browsing. The code is written in Go and uses RPC calls over TCP sockets for communication. Our implementation is composed of 4 different components:

(1) Anonymity Server, which runs the server component of the DC-net and PIR rounds.
(2) Client, which runs the client component of the DC-net and PIR rounds.
(3) Control Server, which coordinates the DC-net and PIR rounds. The control server initiates request rounds when sufficiently many clients are ready. The control server also makes the requests to the outside web, and distributes the responses among the anonymity servers.
(4) Proxy, which handles the proxying of requests and responses from the user's browser.

We implemented the Dissent DC-net which is used for the clients to send their HTTP requests. After the requests are sent, they are mirrored across the anonymity servers and we use our PIR scheme to allow the clients to retrieve their data. Our client can be used as an HTTP proxy, enabling users to anonymously surf the web using our protocol without modification to their browser. We also developed a cryptographically secure PRNG for Go, since one did not exist and we needed it in our implementation of DC-nets.

Our implementation omits some details from the Dissent protocol so we could focus on our PIR scheme. We avoided the verifiable shuffle by assuming that the clients are initialized knowing which DC-net round to use for their broadcast. We didn't implement the blame round because it is only used to remove malicious clients. Our PIR scheme is not vulnerable to interference by malicious clients, it is only vulnerable to interference from malicious servers, which can send incorrect responses. It is straightforward to identify such malicious servers by asking another server to compute the response to the same mask. As such, the complex blame rounds used in DC-nets are not required to protect the PIR component of our protocol.

References

[1] Henry Corrigan-Gibbs and Bryan Ford. Dissent: accountable anonymous group messaging. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 340–350. ACM, 2010.

[2] David Isaac Wolinsky, Henry Corrigan-Gibbs, Bryan Ford, and Aaron Johnson. Dissent in numbers: Making strong anonymity scale. *10th OSDI*, 2012.