A Provably Anonymous Peer-to-Peer Communication Protocol

Henry Corrigan-Gibbs
Bryan Ford, Advisor

February 2, 2010

1 Background

Anonymous communication has a variety of social and political uses. Yet ensuring true anonymity, on the face of it, is completely at odds with network-based communication. The fact that network peers must know the address of their counter-parties makes anonymity in a network context elusive.

A recent paper, by Justin Brickell and Vitaly Shmatikov, describes a solution for the anonymity problem for the domain of data mining [1]. Their solution, which I will refer to as the “Brickov” protocol, provides proven anonymity for the participants and proven data integrity for the data miner. (In the interest of brevity, I omit their definitions of these terms.)

Although Brickell and Shmatikov designed their protocol for use in anonymous data collection, the protocol also applies to the more general case of anonymous communication over a network. The Brickov protocol has two fundamental limitations that keep it from being used for anonymous communication. The first is that any participant can trivially and anonymously disrupt execution of the protocol. The Brickov protocol allows the participants to detect this disruption but not to trace the saboteur.

The second practical limitation is that the Brickov protocol requires each participant to transmit messages of equal length. In a real-world network, participants will want to transmit messages of vastly different lengths. A single user might want to transmit a 100 MB file at a time when other users have no information to transmit. The Brickov protocol is inefficient in this scenario because it would require each participating user to transmit 100 MB of data – even if that data is only padding.

This project will attempt to improve the Brickov protocol to achieve resilience against denial-of-service attacks and to allow for efficient communication of variable-length messages.

Bryan Ford, in collaboration with Justin Brickell and Vitaly Shmatikov, developed the ideas described herein for the extension of the Brickov protocol.
2 Proposed Modifications

2.1 Denial-of-Service Prevention

The first modification will be the addition of a means by which honest participants can identify dishonest participants in the protocol. The most likely approach to addressing denial-of-service attacks will be to allow participants to replay a failed run of the protocol.

As the protocol executes, each participant will periodically broadcast signed traces of the encrypted data she is receiving from and sending to other participants. If the protocol fails, all participants will broadcast their secret session keys. Having every participant’s session keys will allow each user to trace through the execution of the protocol to discover a dishonest participant. (Using this method, only a single dishonest participant can be detected in each run of the protocol.)

If any participant refuses to publish her private session key, that participant will be excluded from the next run of the protocol. If a participant declares that the protocol has failed but the trace finds no dishonesty, that participant will also be excluded from the next run of the protocol.

2.2 Arbitrary Message Length

Modifying the protocol to allow for each participant to broadcast messages of differing lengths is crucial if the protocol is to be used in practice. To achieve this property, participants will use the Brickov protocol as a set-up phase for a “Dining Cryptographers” network (described in [2]).

In the set-up phase, each participant will anonymously submit a fixed-length “slot reservation record” using the Brickov protocol. The slot reservation record will contain information about the data that the participant wants to transmit in the next phase.

The reservation record will include: the message length $L$ and a set of tuples $\{(E_1(s_1), h_1), ..., (E_1(s_n), h_n)\}$ – one for each participant in the protocol. Each tuple contains an encrypted seed $s_i$ for a pseudo-random number generator (PRNG) and a hash of the first $L$ pseudo-random bits generated from $s_i$. The seed will be encrypted with the target participant’s public key and each seed will be generated randomly by the message sender. For her own tuple, the message sender will submit a hash of the data she wishes to send along with a random number as the PRNG seed.

Once the participants generate these reservation records, they will publish them anonymously using the Brickov protocol. The Brickov protocol allows these records to be shuffled anonymously, so that even though each participant will know the length of each message to be sent, the sender will remain anonymous.
To exchange data, the participants will iterate through the slot reservation records. For each record, non-transmitting participants will broadcast a string of random bits using the PRNG seed contained in the record. The transmitter will broadcast her anonymous message XORed with each of the expected PRNG outputs from the other participants. Each participant will then XOR all of these strings of bits together to recover the anonymously sent message.

All participants can verify that every other participant acted honestly by confirming that the expected hash $h_i$ of each participant’s message is equal to the actual hash of that participant’s output.

3 Deliverables

The central product of this project will be a final report describing the modified anonymity-preserving communication protocol. The report will contain a formal definition of the protocol, a description of the protocol’s properties, and proofs of these properties.

Depending on how the project evolves, the deliverables may include an implementation (in C) of the Brickov protocol and some of the aforementioned extensions to the protocol. If an implementation of the protocol does form a core part of the project, the report will include a full description of the implementation’s architecture and design.

References
