CS6490 Project Report: An Awesome Title

Jeff Brock, Roozbeh Gholizadeh, Montgomery Carter

We should look at other protocol papers, to kind of see the normal way of proposing and introducing a new protocol. For example, do we describe what the "protocol does", or do we describe what each "party does" as it performs the protocol?

I know this is shitty writing. I'm just trying to get something on paper. We can figure out what really needs to go in this report, but I thought I would start getting things fleshed out to guide our implementation efforts. I'll clean it up once the protocol is a little better fleshed out. Also, I'm sure much of this "introduction" belongs in the Methodology section.

A separate thought — what if we were to have a service — this REALLY isn't thoguht out — this is just a note to myself. Some sort of service that does SSL to prove to IM clients that it is who it says it is, then sets up key to facilitate DH exchange (this is in contrast to relying on at least one of the IM services to be using SSL to prevent external MitM)

1 Introduction

We rely on large tech organizations more and more to facilitate our daily communication. As a result, these large tech organizations have access to our personal information, from the mundane to the deeply personal and revealing. Blah blah blah.

While operating off the grid may work for some individuals, the convenience of such services provide a compelling argument in favor of their use. The typical attitude is to accept the loss of privacy in exchange for use of the service.

There are many solutions to provide a layer of authentication and encryption that run on top of such services as GMail, Hotmail, Google Hangouts, Skype, etc. However, these solutions are complicated, and generally rely on public-key based authentication to establish shared session keys for encrypting the ensuing communication. Configuring such a solution is complex enough to dissuade the typical user from employing them.

We present the WhizBang Protocol (WBP), a protocol for establishing shared session keys to be used for encryption and integrity protection of instant messaging (IM) communications. However, unlike existing solutions, WBP does not, itself, rely on public key authentication. Instead, WBP relies on the authentication performed by service providers such as Google, Microsoft, or Yahoo, to verify the identities of IM conversation participants. Further, the protocol ensures that service providers do not have access to the shared session keys, and importantly, prevents service providers from employing a man in the middle (MitM) attack, and acting as an intermediary between the two communicating parties.

Unsurprisingly, WBP utilizes a Diffie-Hellman (DH) key exchange for the establishment of shared session keys. The more interesting feature of WBP is its use of two separate messenger services to perform the requisite DH exchange. Each party participating in the IM conversation requires an account with two or more separate IM services.

Each participating party signs on to the initiating <define this "initiating service"> IM service, which authenticates the user with that service. The DH exchange begins by the initiating party sending a message consisting of a DH public key, and a "Next IM Service" field. All parties then sign on to the IM service indicated in the *Next IM Service* field. The recipient parties then generate and send their DH public keys to all participating parties. All parties are now able to calculate the group's DH shared key.

Having established a DH shared key, <I'm making this part up. We need to come up with a real protocol for achieving a shared session secret> each party then sends a nonce, which are then used to generate a shared session key.

Because the initiating party is not listening for a DH key exchange response on the initiating IM service, any MitM response from the server hosting the initiating IM service to the initiating party is disregarded. This prevents this hosting server from performing a MitM. Similarly, because no party is listening for a DH exchange response on the *Next IM Service*, there is no opportunity for the server hosting this second service to perform a MitM attack. Hereafter, we will refer to such an MitM attack by an IM service host which necessarily sits between the communicating parties to relay messages as an *internal MitM attack*.

Still, this leaves an opportunity for a more generic MitM attack, hereafter referred to as an *external MitM attack*. If an adversary has control of part of the route to one of the participants, a MitM attack is still feasible. To address this concern of an external MitM attack, WBP employs two strategies for addressing this vulnerability. The first, and preferred, strategy is to use an IM service that normally performs encryption between the participants and the IM service host for at least one direction of

the DH exchange. This ensures that any intruder on the route will not have access to the plaintext DH key in at least one direction, thus preventing a MitM attack. The second strategy is to exchange a temporary "long term" key between parties, which is then used to encrypt the DH exchange of subsequent conversations between the same parties. However, this second strategy assumes that an intruder was not present during the *initial* conversation between the parties (otherwise an intruder would know the temporary long term key). This temporary "long term" key is replaced after the DH key exchange of each conversation between the parties. This requires an intruder to be present for *every* conversation between the parties, as missing a conversation means the temporary "long term" key they knew has been replaced without them being able to observe the replacement.