Alpenhorn: Bootstrapping Secure Communication without Leaking Metadata

David Lazar and Nickolai Zeldovich

MIT CSAIL

Abstract

Alpenhorn is the first system for establishing session keys between pairs of users, that does not require out-of-band communication and that provides strong *privacy* and *forward secrecy* guarantees for all metadata (i.e., information about who is talking to whom). This addresses a significant shortcoming in all prior works on private messaging, which assume an out-of-band key distribution mechanism.

Alpenhorn builds on two ideas. When a user adds a friend for the first time, Alpenhorn ensures the adversary does not learn the friend's identity, by using *identity-based encryption* in a novel way to privately determine the friend's public key. When starting a conversation, Alpenhorn ensures forward secrecy of metadata by storing pairwise shared secrets in friends' address books, and evolving them over time, using a new *keywheel* construction.

We implemented a prototype of Alpenhorn, and integrated it into the Vuvuzela private messaging system (which did not previously provide privacy or forward secrecy of metadata when initiating conversations). Integrating Alpenhorn into Vuvuzela required changing just 200 lines of code. Experimental results show that Alpenhorn can scale to many users, supporting 10 million users on three Alpenhorn servers with an average dial latency of 150 seconds and a client bandwidth overhead of 3.7 KB/sec.

1 Introduction

The war on privacy is back. Former NSA official Stewart Baker recently said, "Metadata absolutely tells you everything about somebody's life" [8]. Researchers agree [7]. This suggests we will lose the war on privacy if we cannot deploy communication systems that protect metadata—information about who is talking to whom, at what times they communicate, and so on.

Recent research shows that it is possible to build private messaging systems that hide metadata at scale [2–6, 9]. Unfortunately, these systems do not provide users with a convenient way to bootstrap communication without leaking metadata in the process. This impedes practical deployment and precludes any end-to-end metadata privacy guarantees.

Alpenhorn is the first system to address this problem. Functionally, Alpenhorn allows users to initiate a conversation: that is, Alice can use Alpenhorn to call Bob, and Alpenhorn will ensure that Bob knows that Alice is calling, and that both Alice and Bob agree on a fresh cryptographic key, called a session key, to protect their conversation. Alpenhorn is purely a bootstrapping protocol: the actual conversation can take place through one of the systems mentioned earlier. Crucially, Alpenhorn provides *privacy* and *forward secrecy* of metadata. This means that an adversary cannot determine who, if anyone, a user might be calling at any given time, and even if the adversary later compromises a user's computer, they will not be able to tell what calls the user made or received in the past. Much like other key exchange protocols, Alpenhorn also provides authentication to prevent man-in-the-middle attacks.

To understand the challenges faced in achieving Alpenhorn's guarantees, it helps to consider the traditional approach for establishing a session key between users, which works in two steps. In the first step, users learn of each other's long-term public keys, through some public key infrastructure (PKI) system. This step typically corresponds to adding a friend to an address book, which contains the long-term public key of every friend. In the second step, users run a key exchange protocol, such as Diffie-Hellman, to establish a fresh session key, and they use their long-term keys to confirm each other's identity. These two steps correspond to the two challenges faced by Alpenhorn:

First, looking up a user's public key can leak metadata in itself. For instance, if Alice asks a key server for Bob's public key, and the adversary learns about this request, the adversary now knows Alice is about to call Bob. This violates Alpenhorn's goal of achieving *privacy* for metadata, and most existing PKI systems operate in this manner. As a result, it is difficult for a user to populate their address book without leaking information about who they are adding.

Second, even if users somehow manage to obtain each others' public keys, long-term public keys are not a good fit for providing *forward secrecy* for metadata. Specifically, key exchange protocols like Diffie-Hellman authenticate participants by signing messages, which makes it obvious to an adversary who the participants are. A strawman solution is to encrypt these key exchange messages using the other user's public key, and to broadcast these messages, so that an adversary cannot tell who the intended recipient is. However, this fails to provide forward secrecy, because an adversary that later compromises the recipient's computer will obtain that user's long-term private key, and then decrypt those messages to learn about past incoming calls received by that user.

Alpenhorn addresses these challenges using three ideas, as illustrated in Figure 1. First, Alpenhorn uses identity-based encryption (IBE) [1] to achieve privacy for public key lookups. IBE is different from traditional public-key cryptography, in that a server is responsible for generating every user's *private*

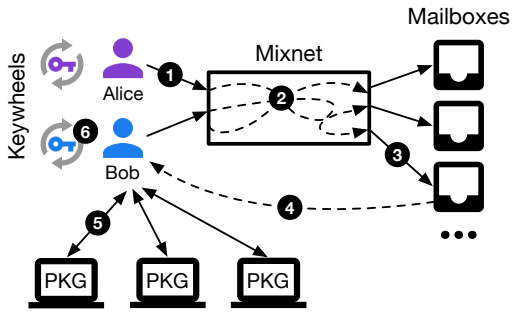

Figure 1: Overview of what happens when Alice adds Bob as a friend using Alpenhorn's add-friend protocol. **1.** Alice's client encrypts a secret using Bob's username as the public key. **2.** The message is sent through a mixnet to hide its source. **3.** The last server in the mixnet groups messages by their destination mailbox. **4.** Bob's client downloads the mailbox that corresponds to his username. **5.** Bob's client proves his identity to the IBE servers to obtain his private key. **6.** For any message that Bob's client can decrypt in the mailbox, the client verifies the source of the message, prompts Bob to accept the friend request, and if so, adds the secret to its keywheel. Now, Alice and Bob have a shared secret, stored in their respective keywheels.

key, which enables the user's *public* key to be a mathematical function of the server's public key and their username (Alpenhorn uses email addresses for this purpose). This allows Alpenhorn to compute a friend's public key without leaking the friend's identity.

Relying on a server to generate the user's private keys is undesirable from a security perspective, so Alpenhorn's second idea is to use IBE only for initially adding a friend to an address book, to use short-lived IBE keys, and to use a set of N IBE servers with an *anytrust* threat model, so that just one IBE server needs to be honest (i.e., properly authenticate the user before giving out the user's private key) to guarantee security. This ensures that, even if some IBE servers are compromised, they cannot violate either the privacy or forward secrecy of metadata for the user's address book operations.

Finally, Alpenhorn must also provide privacy and forward secrecy for the metadata involved in actually initiating a conversation. To do this, instead of storing each other's public keys, Alpenhorn users store pairwise shared secrets in their address books, managed by a novel *keywheel* construction. Alpenhorn's keywheel continuously evolves all shared secrets in a user's address book, so as to provide forward secrecy while still ensuring that, at any given time, two friends have the same secret value in their address books.

Alpenhorn relies on two sets of servers: the IBE servers, mentioned above, and a set of mixnet servers, whose job is to hide the source of every message. Both the IBE and mixnet servers operate in the anytrust model, requiring just one honest server for security. The use of a trusted server allows Alpenhorn to achieve good performance, compared to purely cryptographic approaches like private information retrieval that do not trust any servers at all.

To evaluate Alpenhorn, we implemented a prototype in Go, and integrated it with the Vuvuzela private messaging system, which did not previously provide privacy or forward secrecy for bootstrapping conversations. Integrating Alpenhorn into applications is straightforward; modifying Vuvuzela to use Alpenhorn required changing 200 lines of code. Alpenhorn's performance scales well with the number of users: 3 Alpenhorn servers can support 10 million users with 5% of them initiating a conversation every 5 minutes, with a modest client-side bandwidth cost of 3.7 KB/sec. The client-side overhead in particular is ~6× less than that of the equivalent dialing protocol in Vuvuzela (which also fails to provide the same privacy guarantees).

In summary, the contributions of this work are:

- Alpenhorn, the first system for establishing session keys that provides privacy and forward secrecy for metadata;
- a novel way of using IBE with short-term keys in an anytrust setting to achieve metadata forward secrecy when adding a new friend;
- the keywheel construction, which allows the Alpenhorn client to establish fresh session keys with low latency and low bandwidth overheads;
- a prototype implementation of Alpenhorn, which will be available at `github.com/vuvuzela/alpenhorn`; and
- an experimental evaluation of Alpenhorn that demonstrates it can scale to 10 million users.

References

[1] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *Proc. of the 21st CRYPTO*, Santa Barbara, CA, Aug. 2001.

[2] N. Borisov, G. Danezis, and I. Goldberg. DP5: A private presence service. In *Proc. of the 15th Privacy Enhancing Technologies Symposium*, Philadelphia, PA, June–July 2015.

[3] D. Chaum, F. Javani, A. Kate, A. Krasnova, J. de Ruiter, A. T. Sherman, and D. Das. cMix: Anonymization by high-performance scalable mixing. Cryptology ePrint Archive, Report 2016/008, Jan. 2016.

[4] H. Corrigan-Gibbs, D. Boneh, and D. Mazières. Riposte: An anonymous messaging system handling millions of users. In *Proc. of the 36th IEEE Symposium on Security and Privacy*, San Jose, CA, May 2015.

[5] A. Kwon, D. Lazar, S. Devadas, and B. Ford. Riffle: An efficient communication system with strong anonymity. In *Proc. of the 16th Privacy Enhancing Technologies Symposium*, Darmstadt, Germany, July 2016.

[6] A. Langley. Pond, 2015. `https://pond.imperialviolet.org/`.

[7] J. Mayer and P. Mutchler. MetaPhone: The sensitivity of telephone metadata. `http://webpolicy.org/2014/03/12/metaphone-the-sensitivity-of-telephone-metadata/`, Mar. 2014.

[8] A. Rusbridger. The Snowden leaks and the public. *The New York Review of Books*, Nov. 2013.

[9] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proc. of the 25th SOSP*, Monterey, CA, Oct. 2015.