Pseudo

An Anonymous Email Relay Server

Nicholas Mohr, Johnathon Root, Scott Robinson, Chae Won Lee

May 2015

1 Abstract

Modern secure email protocol relies on PGP encryption to ensure privacy. However, while PGP can confirm message body authenticity and privacy, it fails to provide anonymity. This is because it leaves all identifying headers of an email such as IP and email addresses in plain text. As a result, users are often left vulnerable when communicating with compromised machines. We propose Pseudo, an SMTP relay server capable of stripping identifying information from emails as a solution to improving anonymity when sending emails. Pseudo is intended to be an extra layer of security used on top of PGP to assure not only privacy, but also anonymity. This paper discusses Pseudo's implementation, how it can securely maintain persistent anonymity between two communicating parties, and the shortcomings our implementation faces. A working programmatic implementation can be found at `https://github.com/sdrobs/Pseudo`.

2 Introduction

Email privacy and anonymity are currently major areas of concern among Internet users. Following a number of high-profile government subpoenas for secure email data, it has become clear that it is no longer safe to rely on privacy through PGP encryption alone. This is partly due to the fact that, while PGP succeeds at encryption body text of transmitted messages, the identifying headers of a message such as IP and email addresses are left in plaintext.

Furthermore, even technically secure systems may be compromised as a result of legal obligations. One well-known incident involved Lavabit, an encrypted webmail service founded in 2004 and operated and owned by Ladar Levison.[1] Lavabit used central servers that securely encrypted user information and messages. After rumors that these servers contained information related to the Edward Snowden leaks, the government filed a subpoena forcing Levison to hand over the SSL private keys, compromising all information on the central server. Lavabit's operations were effectively suspended on August 8, 2013.[2]

[1] "Lavabit." 2007. 9 May. 2015

[2] "why I was forced to shut down Lavabit - The Guardian." 2014. 9 May. 2015

In order to address the flaws of existing private email communications, we implemented Pseudo, an SMTP relay server designed to anonymize standard email communication. This server was designed with cases like Lavabit in mind where the data on the server is robust to government breaches. Pseudo ensures email confidentiality by stripping out all identifying information and using randomly generated pseudonyms to identify users talking to each other over email.

Pseudo is a nearly painless layer over standard email, allowing users to reply and send emails as they normally would, while receiving the added benefit of anonymity.

When properly implemented, Pseudo ensures the key idea that no one machine in a network (whether server or client) can give information about another machine on the network if compromised. The basic idea behind this is that only the Pseudo relay server has the ability to determine the proper recipient of a message based on the pseudonym given. This mapping of pseudonyms to email addresses is encrypted at all times with remotely stored keys, and no other email data is ever stored. A user's pseudonym will change based on the conversation and recipient they are conversing with. All other data is wiped after each relay using secure deletion and garbage collection tools.

3 Background

A few different techniques need to be employed to secure the identities of correspondents. Ideally, no plaintext user data can be stored on the server. We use both hashing and encryption to ensure this. The encryption must be secure and unbreakable to protect the identities of the correspondents, and the hash function must also be cryptographically secure (specifically, it must be one-way), as the preimages of the hash are used as encryption keys. We have chosen SHA256 as a hash function and AES as an encryption standard, but these can be easily changed as more advanced standards present themselves. In addition to hashing and encryption, the server naturally receives plaintext data and can store it in the machine's cache or in various logs thanks to the OS, so deleting information stored in this way as soon as possible is necessary. We included various memory-clearing tools implemented in our system to accomplish this. Overall, this ensures that the server is secure in the event of a server seizure or subpoena.

As a result of our security measures, we secure the identities of email correspondents, and do not alter or store the messages sent. This means that, if needed, the correspondents can increase their security by using PGP to encrypt their messages to each other. We use SMTP to send emails, as this is the most popular and accepted standard. This allows users to use Pseudo with popular mail clients like Gmail and Yahoo if they so choose.

We also use a secure data deletion strategy leveraging Unix's shred utility and the BleachBit data deletion application, to make absolutely sure no user data is on the server for any length of time.

4 Protocol

The protocol we developed essentially associates a conversation unique pseudonym with each person's email. This is accomplished by having a trusted third party server organize and encrypt the mapping of emailers to pseudonyms.

4.1 Principals

- User: An individual who wishes to communicate through email without any hard evidence of their identity.
- Trusted Third Party Server: The device that stores pseudonym to email mappings and forwards email to the intended recipient using this mapping.

4.2 Overview

Our protocol has an initialization step and then communication can continue infinitely using this setup. The initial sender sends their email with their subject, message, and their intended recipient's email address. The TTP server receives this email and initializes the pseudonyms for both sender and recipient for this conversation. It then replaces the from field in the email with the pseudonym it generates for the sender. The recipient can now reply to the pseudonym that was sent to them. The server would now look up the matching pseudonym's email and send to that email.

4.3 Pseudonyms

Pseudonyms are used here to increase usability. It's much easier to remember a sequence of words than to remember a sequence of 0s and 1s. This allows users to easily identify and ensure that the person they're speaking to remains the same person that they were speaking to earlier. It also allows them to differentiate between conversations with different people in a reproducible way.

We made the arbitrary choice of using an adjective concatenated with an animal name as the pseudonym. This is easy to remember but the space of pseudonyms is fairly small. This space would be about 2^{32} in size if we allow for 500000 adjectives (the same size as the English dictionary) and 10000 animals. We make this space large enough to be impossible to brute force by concatenating the adjective and animal with a unique id selected randomly from the space of 192 length bitstrings. The adjectives and animal name pairing is also convenient because it is fairly unlikely for there to be collisions that might be confusing within any one person's conversations. By the Birthday bound we have that it will take $7.7 * 10^4$ conversations to have a collision.[3]

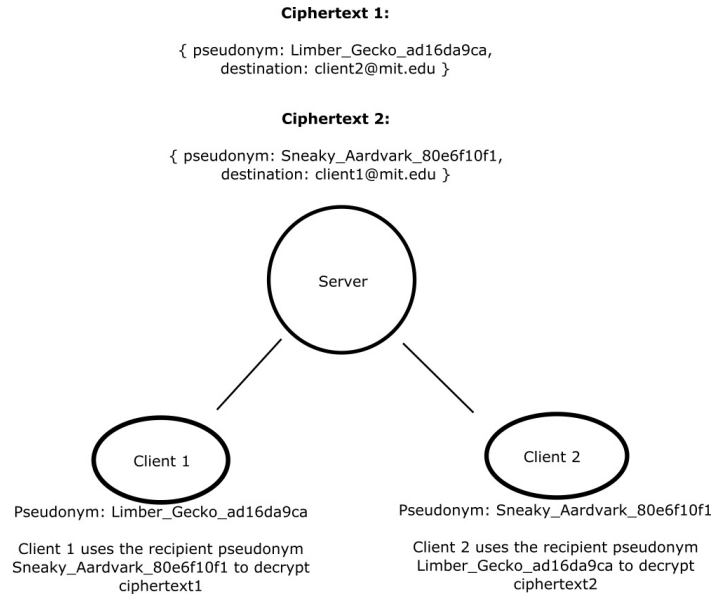

Figure 1: A diagram of the ciphertexts stored by a Pseudo relay server and how client pseudonyms are used to decrypt them

4.4 Conversations

In order to start a conversation between Alice and Bob:

1. Alice sends subject, message, recipient (Bob) to server
2. Server generates pseudonyms for Alice ($Limber\_Gecko\_abcdefabc$) and Bob ($Nasty\_Squirrel\_123456789$) for this conversation, hashes the pseudonyms using SHA256 and encrypts the pseudonym-email pairs using AES.
3. Server sends Alice's pseudonym, subject, message to Bob.
4. Alice deletes the record of her sending a message to Bob from her outbox.

Once this setup has been made all subsequent messages in this scheme can be sent confidentially to the pseudonyms rather than having to be sent to an unencrypted email address. To continue a conversation:

1. Bob replies to Alice's pseudonym with new subject and message. Server verifies Alice's pseudonym by hashing and uses it to decrypt Alice's pseudonym-email pair.

[3] "Birthday Problem" 2015. 10 May. 2015

2. Server sends Bob's pseudonym, new subject and message to Alice.

The server only stores central code, hashes of pseudonyms, and encrypted pairings. No messages, no visible data about email senders or recipients are stored. If Alice or Bob wants the pseudonym-email pairings deleted, they can tell the server to do so. The server will inform Alice and Bob that their conversation has been deleted. No record of the pseudonym-email pairing will remain.

5 Threat Model

Our system was inspired by companies such as Lavabit and Silent circle that cited the impossibility of being unable to maintain confidentiality of its customers' emails should it be served with government orders.[4] [5] We will discuss governmental organizations and malicious individuals in terms of expected behavior and how our system is secure against these actions.

5.1 Government

When an offense under investigation is listed as a violation of the Espionage Act and theft of government property, which was the case for Edward Snowden, the government can make orders intended to monitor a particular user' metadata, defined as "information about each communication sent or received by the account, including the date and time of the communication, the method of communication, and the source and destination of the communication." [6] Our system preserves the anonymity of users, and so the government can only suspect an individual to use Pseudo by rumors. In this case, Pseudo may be expected to be served with a "pen register" which can be obtained without "probable cause" that the target has committed a crime. A Pen-order requires the server operator of Pseudo to "provide all technical assistance necessary to accomplish the installations and use of the pen/trap decide" and to provide the government with the email "from" and "to" lines on every email, as well as the IP address used to access the mailbox. A search warrant takes demands a step further, and asks for all information necessary to decrypt. Since these information are all stripped away when an email is sent by a user, this is not information the government can redeem. Furthermore, because there is no central server the government cannot demand private SSL keys that protect all web traffic as they did for Lavabit.

For Lavabit, the judge rejected Lavabit's motion to unseal the record by stating, "This is an ongoing criminal investigation, and there's no leeway to disclose any information about it."[7] Under a gag order, Pseudo operators may not be able to explicitly state the proceedings of the court or have time to

[4] "Lavabit email service abruptly shut down citing government ..." 2013. 13 May. 2015
[5] "After Lavabit, Silent Circle also shuts down its ... - PC World." 2013. 13 May. 2015
[6] "Edward Snowden's E-Mail Provider Defied FBI ... - Wired." 2014. 13 May. 2015
[7] "Redacted Pleadings Exhibits 1 23 - DocumentCloud." 2013. 13 May. 2015

make a posting under very short notices.[8] However, if Pseudo uses a warrant canary, it can still convey information without the operator taking any action. Unless the government can force the operator to actively post false information, users should be alerted to the fact that the government may be wiretapping or attempting to take information from Pseudo, in which case users can delete their conversations.

5.2 Individual

Our expectations for a malicious individual are more in line with traditional security concerns. Specifically, we are expecting:

- An individual may intercept the server sending a message to a client, and thus have a client's email and the pseudonym of the person they're talking to. Assuming that the individual has no access to the server, the individual only learns that the person is on some pseudo instance independent from the server.
- Even though message bodies are encrypted with PGP, the first email is still insecure, as the header contains the identity of the recipient.
- Since the from/to can take in any arbitrary string, a malicious individual may inject javascript in the from/to and execute arbitrary code. This is why we sanitize our inputs in our protocol, which is fairly standard.

6 Security

The entire concept of our system is based upon keeping any party from being a single point of failure. We accomplish this by giving each part of the system as little information as possible.

6.1 Clients

Because we do not control the client's machine, we cannot control the level of security on it. We can only identify the consequences of a compromised client machine. The client's information about where any emails came from is not hashed or encrypted. However, this information is entirely under pseudonyms, with the exception of the initial email sent to begin a conversation. A Pseudonym also does not reveal the identity of the sender since they are chosen at random for each conversation. Since the pseudonyms are specific to conversations, knowledge of a pseudonym does not unmask someone who has conversed with a compromised machine. These pseudonyms allow the user to verify which messages belong in the same conversation. This could give the adversary enough information to implicate someone if their emails were explicit

[8] "Lavabit founder, under gag order, speaks out about ..." 2013. 13 May. 2015

enough. In addition, knowing a pseudonym allows the adversary to impersonate one of the people in the conversation. This is acceptable; impersonation of any email account is possible under SMTP without limits. Impersonation in a pseudo conversation requires a receiving pseudonym. We expect our users to delete the initial email they send, as it would endanger the person they are communicating with since the "to:" field is unencrypted.

6.2 Server

The server's permanent information is either hashed or encrypted with a key. Our hash-function must be one-way in order to prevent those who might compromise the server from dehashing the pseudonyms. We assume that SHA256 is one-way. Our encryption must be infeasible to decrypt without the key. We are using AES without modification and so we assume this property. Because we use the pseudonyms as keys and we rely on the inability to find their hashes the keys must also be computationally infeasible to guess. We solve this by relying on our random unique id at the end of the pseudonym to give us 192 bits of entropy. This means that we require our random number generator to reliably choose random numbers. Overall, the stored data is secure.

All transient information on the server such as message contents and the plaintext pseudonyms is deleted once the message has been sent. We delete this data using the UNIX command shred. Using shred, we overwrite the file with random data 5 times, overwrite that random data with zeros, truncate and finally delete the file. This allows us to securely delete the emails the server receives. We also shred files in case of a In addition to main memory, the server's cache may hold on to message data. BleachBit is a program that can securely delete many parts of a system's memory; we use this in order to clear the cache whenever an email is processed. Unallocated space may also hold message details for longer than we would like. We will clear unallocated space on a scheduled interval in order to minimize the downtime on the server since clearing unallocated space is an operation that takes a long time.

7 Vulnerabilities

While Pseudo does succeed at building an extra layer of security upon PGP, there are some shortcomings in the implementation of such a system that do not allow us to easily solve them.

Perhaps the most pressing issue is that of 'sent' receipts on a user's computer. The user who initiates an email conversation with pseudo will need to specify the recipient's email in plaintext in the first message of any conversation. While this initial sender has no worries about their email being displayed on the end-client's computer, they will in fact retain a receipt of who they are conversing with in their 'sent' folder if not careful. This is worth noting: the initiator of a conversation is guaranteed anonymity on the recipient's computer, but the responder is not.

Another concern in this scheme is the idea of a persistent man-in-the-middle attack. As the server and the clients need to communicate to interact with each other, it is critical that communication between all parties in this scheme is secure. If a man-in-the-middle attack is possible, an adversary can accomplish a few different things. If the initial message in a conversation is intercepted, then, just as if the sender did not delete the message from their outbox, the adversary knows both of the correspondents in the conversation. If another message in the conversation is received (ex. from Limber Gecko to B), then the adversary does not know the identity of Limber Gecko, but they do know the identity of the Pseudo server sending the email, which could be a risk to the individual if using the server itself is incriminating or of interest. While a man-in-the-middle attack is a very real possibility we feel that it would be an unlikely occurrence that the first email sent to pseudo would be intercepted. This vulnerability already exists from SMTP and so Pseudo has only increased the security of the identities of correspondents in this regard.

The server needs to operate on plaintext email addresses over the course of its operation to relay emails to its clients. Due to this, it is entirely possible for the server to be seized or otherwise compromised and have its data read while the plaintext emails were in the server's cache. This compromises the identity of clients using the system, as plaintext pseudonyms or emails reveal the server's users, but does not necessarily compromise the messages or conversations themselves, which can be protected by PGP.

8 Conclusion

Pseudo allows users of email to better address their security needs by hiding the correspondents in an email conversation. In conjunction with other security tools, it allows for secure communication in an insecure world. However, improvements to the system are possible. In the current implementation, security is somewhat compromised for the ability to use a Pseudo server in conjunction with any email client. Subsequent versions of Pseudo could be improved by making a special client program just for using Pseudo. This client would encrypt the initial message to the server with the server's public key—meaning that it would be impossible to man-in-the-middle the system for header detail. Having a client program would mean that the process of writing and securing an email would not be upon the client, they would just be using it as if they were writing any other email. These versions with a specialized client would be much tougher to get people to adopt but would offer greater security and it would be much easier to attain maximal security.

9 References

1. `https://lavabit.com/`

2. http://www.theguardian.com/commentisfree/2014/may/20/why-did-lavabit-shut-down-snowden-email

3. http://en.wikipedia.org/wiki/Birthday_problem

4. http://www.theguardian.com/technology/2013/aug/08/lavabit-email-shut-down-edward-snowden

5. http://www.pcworld.com/article/2046264/after-lavabit-silent-circle-also-shuts-down-email-service.html

6. http://www.wired.com/2013/10/lavabit_unsealed/

7. https://www.documentcloud.org/documents/801182-redacted-pleadings-exhibits-1-23.html

8. http://arstechnica.com/tech-policy/2013/08/lavabit-founder-under-gag-order-speaks-out-about-shut-down-decision/