demonsaw

Demonsaw is a new type of message and data transfer application that's secure, anonymous, free, and everywhere. It's designed to protect our anonymity and hide what we're sharing. Unlike traditional P2P networks, our IP addresses are never revealed to the world. Demonsaw is wrapped in multiple layers of user-derived cryptographic algorithms based on a new and modern security model called, Social Encryption. We can share whatever we want, with whomever we want, without fear or consequences. Safeguarding our privacy and protecting our Right to Share are the primary goals of demonsaw.

A History of Data Sharing

Every data sharing application in the past 45+ years has had 3 common elements: the ability to share data, the ability to control program flow, and the ability to transfer data. Demonsaw is different from other data sharing applications because it breaks up these different abilities into separate network components that act independently. By doing this, demonsaw creates secure layers of abstraction that completely secures our data sharing.

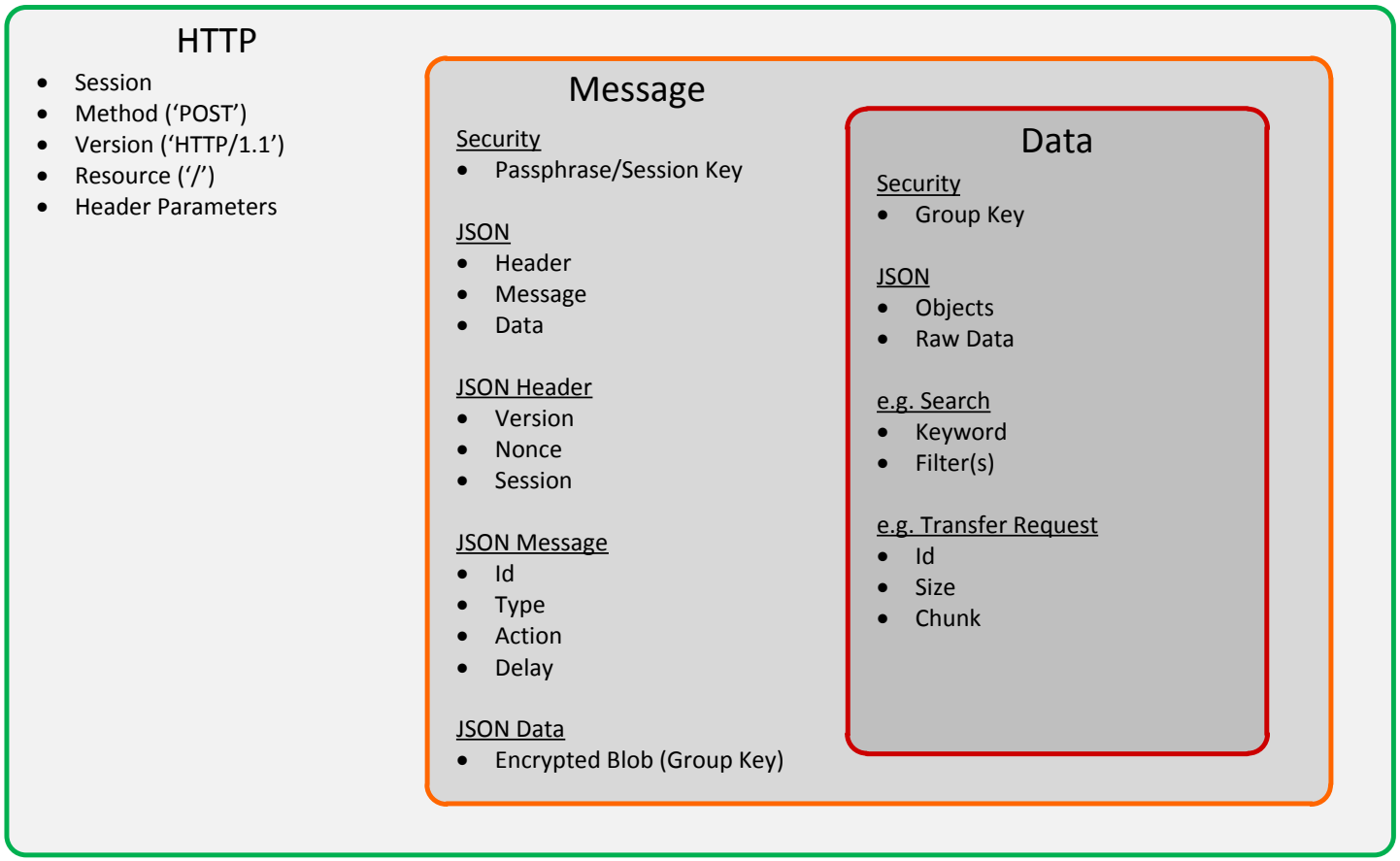
How Does It Work?

Demonsaw encrypts all messages in multiple layers of crypto. All encryption keys used within demonsaw are generated at runtime and never sent over the network. This means that clients on a demonsaw network only have to trust themselves – there are no network nodes with "authority" that can be compromised by corporate or government agencies.

A demonsaw network is defined by multiple clients and router(s). A client can belong to 1:N groups, each of which has a unique cryptographic signature that all members of the group must possess. Social Encryption is "the art of hiding our secrets within the fabric of social interaction", and allows clients of a given group to derive the same encryption keys by shared and contextual knowledge. In demonsaw all clients of a given group will derive the same encryption keys because they all reference the same image on their computer. In demonsaw 2.0 this security model is extended to any number of locally or online accessible URI's.

Routers on a demonsaw network can either route messages, data, or both. Message routers perform very light-weight client synchronization duties and are responsible for grouping clients and managing their session lifetimes. Message routers know nothing about what data is being shared, transferred, etc. Data routers provide the conduits through which clients will upload or download data packets in real-time. Data routers never cache data chunks nor even know what the chunks of data contain – like message routers, data routers are completely oblivious to the underlying contents of the data being transferred.

This model of message/data separation within demonsaw is very intentional and results in a layered security model that prevents any node in the network from being able to discern what a client is transferring. By physically and logically separating the message and data routing components we achieve a heightened level of security not found in other secure message and data transfer networks. In fact, not even clients within the same group are able to know who is sharing what.



Controlling Program Flow

Message routers are the initial contact points for all clients who want to participate in a given group (a.k.a. demonsaw network). Message routers group clients, manage sessions, and control program flow. Message routers have no visibility into what the clients in the network are doing, including data transfers. Even if the message router were to be compromised by a rogue agency, it would still be blind to the client-to-client communications unless it also compromised one of the clients as well. By using HTTPS, instead of HTTP as our message conduit, we can even prevent Deep Packet Inspection (DPI) attacks. Note that HTTPS support is one of the many features on the demonsaw roadmap.

Message routers can be hosted on homes or offices to help minimize the chance of compromise by rogue agencies. By design, message routers can run on very low-power devices, like the Raspberry Pi. Being able to host an entire secure message and data transfer network on a $35 computer is very appealing to lot of people.

In demonsaw 2.0 (available in 3 months) message routers will be able to trust each other, meaning that groups can be aggregated cross-routers to create very large universal groups. This will provide a fault-tolerant network in the event that some of the message routers are shutdown. Imagine what appears to be a demonsaw network of 10 million clients that's aggregated across 10,000 Raspberry Pi devices spread throughout the world. The potential for such a secure message and data transfer scenario is unbounded.

Data Transfer

Message routers can be configured to route clients to 1:N data routers.

Data routers receive data from uploading clients and hand these out to downloading clients. There is no direct interaction between the uploaders and downloaders. By separating the physical devices that perform message and data routing, we achieve a heightened degree of security not found in any other secure message and data transfer network. This architectural separation also allows us to scale horizontally to meet the emerging performance needs of the network. We can independently add new data routers irrespective of the total number of clients attached to the network. Like message routers, data routers can be added or removed from the network without compromising the integrity or availability of the network. This gives us flexibility and natural fault tolerance to resist attack from rogue agencies.

Sharing Data

There is no P2P in demonsaw, nor is there any client-server interaction. P2P interaction is very dangerous, and is a primary reason for the degradation of privacy and lack of protection in programs like uTorrent. And client-server architectures are problematic with respect to logging, hacking, backdoors, and 3rd parties unwillingly abusing our data.

When we share data in demonsaw, we're actually not sharing data with anybody. Instead, we're uploading or downloading data through 1:N data routers that can be spread all over the world. Since data routers are separate from message routers, it's impossible to compromise a message or data router and be able to discern the content of the data being transferred. The multiple layers of encryption in demonsaw also aids in this effort. Data routers can be hosted in at traditional web hosting sites for $3-4/month. This is exactly what I do. Last month I sent ~50GB of data through a hosting account that costs me $3 per month. This is one of the key benefits of the demonsaw architecture.

Program Flow Example

In the following example, there are 2 groups ("demonsaw networks"), 0xEFF and 0x0FF. R_0 and R_1 are message routers that are hosted privately on someone's home network.

Clients C_0, C_1, C_2, C_3, C4, and C_5 are members of 0xEFF, meaning that they share the same encryption keys derived from a common image. It is assumed that previous contextual knowledge existed thereby enabled these 6 clients to know which image was to be used for the creation of the encryption keys that allowed them to join the same group.

Clients C_6 and C_7 are members of the group 0x0FF.

Only clients in the same group can search, browse, and transfer data. This is due to the nature of the cryptography and the fact that all client-level data is encrypted with the AES key derived from the shared group image. In fact, a demonsaw group is defined by the unique set of bytes that make up a specific image. To change the image is to change the group.

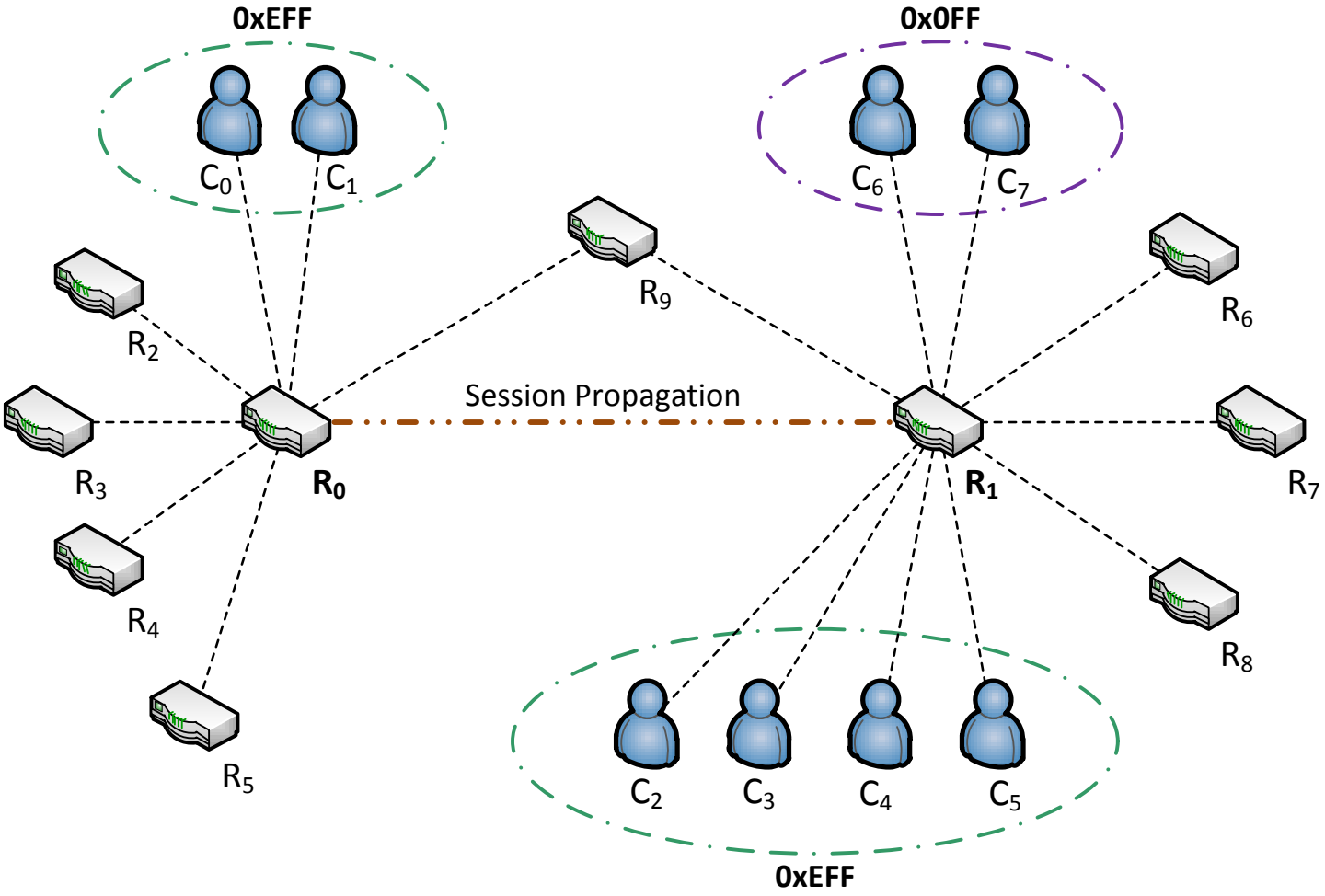
R_2, R_3, R_4, R_5, R_6, R_7, R_8, and R_9 are routers that are configured to only route data (not messages). The message router, R_0, is configured to route client uploads/downloads to R_2, R_3, R_4, R_5, and R_9, while R_1 is configured to route client uploads/downloads to R_6, R_7, R_8, and R_9.

Let's assume that C_2 searches the network for "ACDC". Since C_2 is a member of group 0xEFF it will only search clients within 0xEFF and not any other groups. When C_2 submits its search it encrypts the search details (i.e. keywords and filters) and sends this request to R_1. R_1 doesn't know what C_2 is searching for – all it knows is that a search request in

0xEFF has been issued. R_1 then forwards the search request to the other members of 0xEFF, who each in turn decrypt the underlying search request and responds with an HTTP OK or NOT_FOUND, indicating whether they have any results or not. If they have search results they will encrypt them with the group key and forward these back to R_1 with the HTTP OK response. Note that this is all an asynchronous process and fairly instantaneous. For the sake of this example, let's assume that C_0 has the file "ACDC – Highway to Hell".

C_2 receives the search results back from R_1 and decrypts them to learn that somewhere on the network a client has "ACDC – Highway to Hell". C_2 encrypts the request to download this file and sends this to R_1. R_1 doesn't know what C_2 is downloading – all it knows is that a download request in 0xEFF has been issued. R_1 then forwards the download request, along with which data router to go to (R_9 in this example), to the other members of 0xEFF, who each in turn decrypt the underlying download request and responds with an HTTP OK or NOT_FOUND, indicating whether they have any results or not.

When C_0 receives the transfer request it connects to R_9 and begins to upload "ACDC – Highway to Hell". C_0 encrypts all chunks of the file with the group key, ensuring that the R_9 won't be able to discern what the data is. C_5 then connects to R_9 and starts downloading and decrypting the chunks of data. Once all of the chunks of "ACDC – Highway to Hell" are downloaded, the operation is complete.



Demonsaw 2.0

I just released version 1.5 on May 16th, 2015. This version was a huge step forward, involving a port from C# to C++ code as well as a release on Linux and Raspberry Pi. In three months I will be releasing version 2.0, which will hopefully include all of the following features:

- Secure message (chat)
- Auto-sync files/folders (like Dropbox)
- Session Propagation (aggregate, cross-router groups)
- Windows, Linux, OSX, Raspberry Pi, Android router/client GUI and command-line interfaces
- Instantaneous downloads and multi-threaded transfers
- Streaming
- HTTPS

Summary

Demonsaw provides clear and concise lines of separation between the sharing of data, the controlling of program flow, and the transferring of data. This is the key ingredient that gives demonsaw the ability to secure and anonymize our secure message and data transfers. By privatizing the message routers and abstracting the data routers, demonsaw gives clients a naturally fault tolerant and scalable architecture in which to transfer message and data securely and anonymously.

It is my hope that demonsaw gives us an advantage in our fight against oppressive governments and corporations who continue to support legislation and illegal practices designed to infringe upon our most basic and fundamental rights to privacy, secrecy, and personal beliefs. Thank you.

Eijah
eijah@demonsaw.com