Survey of Mesh Networking Messengers

Simon Blöchinger, Richard von Seck*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: simon.bloechinger@tum.de, seck@net.in.tum.de

***Abstract*—A centralized architecture utilizing one or more central servers is used by most messenger applications. The messenger will only work if the server is functioning and a connection is possible. Mesh networking messengers use peer-to-peer connections to exchange messages directly, without the need for central servers.**

A decentralized architecture is more resilient against failures. Mesh networking messengers have privacy benefits as well. This paper analyzes different mesh networking messengers and compares their features.

Briar and Technitium Mesh provide secure mesh networking messaging for their respective platforms Android and Windows. Meshenger implements encrypted local audio and video calls.

***Index Terms*—mesh networking messenger, mesh messenger, peer-to-peer messenger, mesh network, peer-to-peer network, instant messaging**

1. Introduction

Most messengers use central servers responsible for storing and exchanging messages. These messengers only work when they have access to the Internet and the central servers are available. In situations where the users cannot connect to the internet, for example in remote locations or when the necessary infrastructure fails, the messenger cannot function. The same is true when the central servers are not available. Even if a direct connection between the clients would be possible, messengers relying on central servers do not work without them.

A connection between peers is called *peer-to-peer*. As shown by Akyildiz et al. in [1], when multiple peers are dynamically interconnected peer-to-peer, this is called a mesh network. Mesh networks often allow routing through the client nodes. Messengers that utilize the mesh networking approach are called mesh networking messengers. These messengers do not rely on central servers but connect to each other directly.

The decentralized mesh architecture does not have a single point of failure. As long as there are enough redundant connections between the devices forming the mesh, no device is essential.

Another advantage of mesh networking messengers is that they are privacy friendly. When there is no central server storing the messages, central data mining is not possible. There is also no chance of server data leaks. Vulnerabilities in the messaging application itself can still exist.

Because mesh networking messengers are inherently attractive to people interested in privacy-oriented messaging, most mesh networking messengers are open-source as well. This allows users to inspect the code of the application they are using themselves to make sure that there are no hidden side effects.

In Section 2 three types of mesh networking systems are introduced and use cases explored.

In Section 3 different mesh networking messenger applications are analyzed and compared.

2. Mesh Based Networks

There are three different types of mesh networking systems [1]. The *Infrastructure Mesh Network* is differentiating between infrastructure and clients. The infrastructure is interconnected in a mesh, the clients are connecting to the infrastructure.

The *Client Mesh Network* only has a single type of node, the client. All clients are connected and pose not only as an end-user device but can also be used to route messages. The nodes in this network are communicating using peer-to-peer connections.

The *Hybrid Mesh Network* combines the infrastructure and the client approach. The clients can access the network both through routers, which make up the infrastructure, and through the other clients.

2.1. Use Cases

Infrastructure Mesh Networks can be used to set up routers on a large scale to provide a connection to the Internet in an area that is too big for a single router. The area is set up with multiple routers in such a way that allows every part of the area to be reached by at least one router. Then the routers automatically and dynamically form a mesh network and route messages between them [1].

As presented by Coulouris et al. in [2], Hybrid Mesh Networks are great for hosting big, immutable files such as video files on a large scale. Since the files are immutable, they can be stored in small parts across a distributed network without worrying about keeping them up to date. For downloading purposes, the parts can be supplied by multiple hosts. This makes a bandwidth problem on the hosters side less likely. After the download, the downloader can become a hoster on his own. This can help to balance out supply and demand for a file.

doi: 10.2313/NET-2021-05-1_01

3. Mesh Networking Messengers

The focus of this paper is the comparison of different mesh networking messengers. The features of Briar and Technitium Mesh are evaluated in depth. Also considered are Meshenger, Serval Mesh/Chat, FireChat and Bridgefy.

3.1. Briar

Briar is an open-source messenger with a strong focus on privacy that uses a mesh networking approach which allows users to privately communicate with each other. It was first released in 2018 for Android [3]. The devices can connect anonymously over the Internet via Tor or locally via Wi-Fi or Bluetooth.

All direct communication using Briar can only happen between contacts. There is no possibility to send messages directly to a non-contact.

3.1.1. Adding Contacts. A nearby contact can be added by exchanging QR codes. Using an already existing communication channel, a contact can be added by exchanging a link. If two users share a common contact, they can be introduced to each other via this common contact.

Ways of reaching each contact are stored locally on the user's device. If a connection via multiple transport mediums is available, they will be used in parallel.

When adding a contact by exchanging QR codes, the *Bramble QR Code Protocol* (BQP) is used [4]. When adding a contact at a distance by exchanging a link or by introduction through a common contact, the *Bramble Rendezvous Protocol* (BRP) is used [5].

Both protocols are similar and serve the same purpose: an initial public key exchange is used for authentication and encryption. As a result of the protocol, both users know how to reach their contact and have a shared secret key. This shared secret key is used to derive other keys from it, which then are used to encrypt the communication. Both protocols use the Diffie Hellman key agreement function and are secure, even if an attacker can read, modify, delete and insert traffic on all transports at will, as long as the initial public key exchange is not modified.

During the BQP, a commitment to a public key and information on how to be reached using Bluetooth and Wi-Fi, the short-range transports that are supported by Briar, is shared using a QR code. The participants establish an insecure connection and share ephemeral public keys. Using these public keys, a secure connection gets established. Then the participants agree on a shared master key via the secure connection.

At the beginning of the BRP, the only information known to each participant is the public key of the other. Both parties generate pseudo-random contact details for themselves and the other's endpoint using the shared public keys. A shared secret key is also derived from the public keys. For the next 48 hours both peers listen on their network endpoints. If no connection happens within 48 hours, the rendezvous is considered to have failed. If a connection can be established, the participants exchange long-term contact details.

3.1.2. Methods of Communication. Briar allows for a few different methods of communication between hosts: *private chats*, *private groups*, *forums* and *blogs* [6]. Each method of communication can use any of the transport mediums that are available.

Private chats allow users to chat with one of their contacts.

Private groups are created by one user. This user is the owner of the private group. Only the owner can add his own contacts. If the owner leaves the private group, the private group will be dissolved.

Forums are similar to private groups with the exception that every participant is equal. Everyone can invite their contacts and the forum will not be dissolved if the original creator leaves.

It is possible for two users who are not each other's contacts to be part of the same private group or forum.

The blog behaves similar to a broadcast to all contacts. Anything that gets published in a blog can be read by and commented on from all contacts.

New messages in private groups and forums are shared with all contacts that are in the same private group or forum. This allows private group or forum updates to spread to people without a direct connection to the sender of the update. Note that this sharing only ever happens with contacts. No sharing happens to users who are in the private group or forum, but not a contact.

To receive a message, the sender and the receiver need to be connected with each other. Sending a message that can get received even when the sender is offline is not possible. If the receiver is not online when a message gets sent, the sender periodically tries again until the message was successfully sent and received. Briar uses a background task to send and receive messages.

3.1.3. Bramble Transport Protocol. The transport of data between two parties in the Briar application is done using the *Bramble Transport Protocol* (BTP), which provides a secure channel ensuring confidentiality, integrity, authenticity and forward secrecy across a wide range of underlying transports. The protocol is difficult to distinguish from other protocols. To further hide the use of Briar, techniques like traffic morphing can be used. It is suitable for delay-tolerant networks and can even be used on transports with very high latency, such as sending a physical storage medium through mail. The BTP uses rotating keys to encrypt and decrypt the message stream. [7].

3.1.4. Conclusion. Briar implements secure mesh text messaging for Android. It is able to establish encrypted connections via Wi-Fi, Bluetooth and via the Internet using Tor. Briar is open-source. It is of limited usefulness when it comes to communicating with a group of local strangers in case of infrastructure failure, because communication can only happen between contacts.

3.2. Technitium Mesh

The mesh networking messenger Mesh by Technitium is another messenger that provides peer-to-peer communication. Its alpha version was released in 2019 for Windows. Text messaging and file transfers are possible. It is a

doi: 10.2313/NET-2021-05-1_01

direct successor to the Bit Chat project. Most of Bit Chat's design is found in Mesh as well [8], [9].

3.2.1. Technitium Bit Chat. The concept for Bit Chat was invented in 2011. It takes many of BitTorrent's concepts with the goal of instant messaging instead of file sharing. Connections are made using existing BitTorrent trackers, which are centralized servers providing information about the location of files.

Instead of the location of files, the trackers are storing who is part of which channel. This is done by storing the IP addresses of all participants together with a unique infohash that identifies the channel. After receiving the IP addresses of the other participants, a direct authenticated connection can be established via IP. Public key cryptography is used to achieve authentication and confidentiality [10].

Bit Chat requires a central user profile registration based on email. Mesh does not use central user profiles. Instead users are identified with a user ID generated from their RSA key pair.

Mesh also removes the BitTorrent trackers and replaces them with *Distributed Hash Tables* (DHT). Using the BitTorrent trackers can lead to connectivity problems since some ISPs block BitTorrent traffic.

Mesh users can choose to use an *anonymous profile* instead of a *peer-to-peer profile*. Anonymous profiles use Tor onion services to accept inbound requests. For every login a new onion domain name is used to prevent tracking. Communication between anonymous and peer-to-peer profiles is possible. Connections using an anonymous profile are still peer-to-peer connections [9].

3.2.2. Methods of Communication. Mesh provides two different options for communication: private chat and group chat [11].

Using the user ID and an optional password, a private chat can be initiated. These need to be transmitted using a secure channel not provided by Mesh.

To initiate a group chat, a group name and an optional password is needed. The name and password have to be distributed to participants. This can be done through private chat or any other secure external channel.

3.2.3. Protocols. Mesh uses the symmetric-key algorithm AES-256. To share the key with all participants, the Diffie-Hellman key exchange function is used. During the key exchange, the user IDs of the participants are verified using RSA. To provide perfect forward secrecy, a new key exchange is done periodically. Message authenticity is ensured through the use of *HMAC-SHA256*. The local data stored on the user's devices is encrypted using a secure key derived from the user's password by the *Password-Based Key Derivation Function 2* (PBKDF2). In Mesh's implementation, PBKDF2 uses the pseudorandom function HMAC-SHA256.

When a new channel is created, the network ID of the channel is used to uniquely identify the channel. In a group chat, the network ID is the hash of the group name in combination with the group password. In a private chat, it is the hash of the user IDs of the participants in combination with the password. This hash is then stored together with the IP addresses of the channel participants in a *Distributed Hash Table* [11].

3.2.4. Conclusion. Technitium Mesh implements encrypted peer-to-peer text messaging and file transfer in pairs and in groups. Communication is possible locally via LAN/Wi-Fi and globally using IP or using Tor onion services. Mesh is released on Windows and is open-source. Mesh does not implement mesh network routing functionality for messaging.

3.3. Other Messengers

While Briar and Technitium Mesh are analyzed in detail, other messengers are considered as well.

3.3.1. Meshenger. Meshenger is an open-source peer-to-peer messenger for audio and video communication released on Android. The project started in 2018 as part of the Freifunk initiative [12]. Version 1.0 got released in 2018 [13], followed by a repository change [14].

Meshenger supports encrypted audio and video calls in local networks with contacts. Text messaging is not supported. Communication via Bluetooth or via the Internet is also not supported.

To establish a connection with a contact, primarily local unicast IPv6 addresses are used. Other IP addresses or DNS names can be used as well. Meshenger does not use mesh routing for audio and video calls.

3.3.2. Serval Mesh/Chat. The Serval Project has the goal to help the geographically, financially or otherwise unfortunate.

Serval Mesh is an open-source Android app that provides secure mesh networking text messaging, file sharing and audio calls using Wi-Fi. Audio calls only work under good conditions. Group or broadcast messaging is not supported.

Serval Chat is an iOS app providing secure text messaging using Apple's proprietary peer-to-peer wireless network. Group and broadcast messaging are supported. Serval Chat is not open-source [15].

Communication between Serval Mesh and Serval Chat is not possible. While the project in general has interesting and unique features, there has been no development since 2018 for Serval Mesh/Chat. Serval Mesh is not available in Google Play anymore and Serval Chat is also not available in Apple App Store [16].

3.3.3. FireChat. FireChat was a mesh networking messenger that got popular during protests in Iraq and Hong Kong in 2014 [17], [18]. It has since been discontinued and the official website is not available anymore [19].

3.3.4. Bridgefy. Bridgefy is another mesh networking messenger. It is released for Android and iOS [20]. A Bluetooth connection is used to connect the devices. It got used in Hong Kong in 2019 [21].

While there are code samples for developers, Bridgefy is not open source [22]. Bridgefy currently still has security issues and is not able to provide secure messaging [23].

doi: 10.2313/NET-2021-05-1_01

	Briar	Mesh	Meshenger	Serval Mesh/Chat	Bridgefy
Communication	Contacts	Chat rooms	Contacts	Private & broadcast (iOS)	Private & broadcast
Bluetooth	Yes	No	No	No	Yes
LAN / Wi-Fi	Yes	Yes	Yes	Yes (Android)	Yes
IP	No	Yes	Yes	No	No
Tor	Yes	Yes	No	No	No
Built secure	Yes	Yes	Yes	Yes	No
Platform	Android	Windows	Android	Android, iOS	Android, iOS
Open-source	Yes	Yes	Yes	Yes (Android)	No
Text	Yes	Yes	No	Yes	Yes
Audio	No	No	Yes	Yes (Android, limited)	No
Video	No	No	Yes	No	No
File sharing	No	Yes	No	Yes (Android)	No

Figure 1: Mesh networking messengers - comparison

4. Conclusion

Briar and Technitium Mesh implement secure mesh networking text messaging for their respective platforms Android and Windows. They implement encrypted peer-to-peer communication both over local and global transport mediums.

Briar only allows communication with contacts, which limits its usefulness in communicating with local strangers, for example in case of a local infrastructure failure.

Mesh allows to connect to an open local LAN chatroom without a password. File sharing is also possible. Mesh does not implement mesh network routing functionality for messaging.

Meshenger allows for secure audio and video communication in local networks. Meshenger also does not implement mesh networking functionality for audio and video calls.

A big weakness of these mesh networking messengers is that they are only able to communicate with other devices using the same application. Since there is a variety of mesh networking messengers that come and go, a messenger is not of great use if there are not enough users.

If a standard would get introduced for mesh networking messaging, the usefulness of these mesh networking messengers might rise. But because these messengers use different protocols and have different design goals, it is unlikely that messengers supporting inter-messenger communication will become common.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Computer networks*, vol. 47, no. 4, pp. 445–487, 2005.

[2] G. F. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*. pearson education, 2005.

[3] M. Rogers, E. Saitta, T. Grote, J. Dehm, B. Wieder, and N. Alt, "Briar Release," https://briarproject.org/news/2018-1.0-released-new-funding/, 2018, [Online; accessed 03-October-2020].

[4] ——, "Bramble QR Code Protocol," https://code.briarproject.org/briar/briar-spec/blob/master/protocols/BQP.md, 2019, [Online; accessed 21-November-2020].

[5] ——, "Bramble Rendezvous Protocol," https://code.briarproject.org/briar/briar-spec/blob/master/protocols/BRP.md, 2019, [Online; accessed 21-November-2020].

[6] ——, "Briar Manual," https://briarproject.org/manual/, 2016, [Online; accessed 03-October-2020].

[7] ——, "Bramble Transport Protocol," https://code.briarproject.org/briar/briar-spec/blob/master/protocols/BTP.md, 2019, [Online; accessed 21-November-2020].

[8] S. Zare, "Technetium Mesh," https://mesh.im/, 2019, [Online; accessed 21-November-2020].

[9] ——, "Technetium Mesh Release," https://blog.technitium.com/2019/12/technitium-mesh-released.html, 2019, [Online; accessed 03-October-2020].

[10] ——, "Technetium Bit Chat Release," https://blog.technitium.com/2011/07/bitchat-peer-to-peer-instant-messaging.html, 2011, [Online; accessed 03-October-2020].

[11] ——, "Technetium FAQ," https://mesh.im/faq.html#q15, 2019, [Online; accessed 21-November-2020].

[12] D. Dakhno, "Meshenger Original Repository," https://github.com/dakhnod/Meshenger, 2018, [Online; accessed 03-October-2020].

[13] ——, "Meshenger - P2P Local Network Messenger - Final Update," https://blog.freifunk.net/2018/08/14/meshenger-p2p-local-network-messenger-final-update/, 2018, [Online; accessed 03-October-2020].

[14] ——, "Meshenger 2.0 Repository," https://github.com/meshenger-app/meshenger-android, 2018, [Online; accessed 03-October-2020].

[15] P. Gardner-Stephen, "The Serval Project," http://servalproject.org/, 2011, [Online; accessed 03-October-2020].

[16] ——, "Serval Project Blog," https://servalpaul.blogspot.com/, 2011, [Online; accessed 03-October-2020].

[17] A. Hern, "Firechat Updates as 40,000 Iraquis Download 'Mesh' Chat App in Censored Baghdad," https://www.theguardian.com/technology/2014/jun/24/firechat-updates-as-40000-iraqis-download-mesh-chat-app-to-get-online-in-censored-baghdad, 2014, [Online; accessed 03-October-2020].

[18] A. Bland, "FireChat - The Messaging App That's Powering the Hong Kong Protests," https://www.theguardian.com/world/2014/sep/29/firechat-messaging-app-powering-hong-kong-protests, 2014, [Online; accessed 03-October-2020].

[19] O. Garden, "FireChat," https://www.opengarden.com/firechat, 2014, [Online; accessed 03-October-2020; not available].

[20] "Bridgefy," https://bridgefy.me/, 2020, [Online; accessed 03-October-2020].

[21] J. Koetsier, "Hong Kong Protestors Using Mesh Messaging App China Can't Block: Usage Up 3685%," https://www.forbes.com/sites/johnkoetsier/2019/09/02/hong-kong-protestors-using-mesh-messaging-app-china-cant-block-usage-up-3685/, 2019, [Online; accessed 03-October-2020].

[22] "Bridgefy Code Samples," https://github.com/bridgefy, [Online; accessed 03-October-2020].

[23] "Bridgefy's Commitment to Privacy and Security," https://bridgefy.me/bridgefys-commitment-to-privacy-and-security/, 2020, [Online; accessed 03-October-2020].