Matrix Cryptography

Franziska Steinle, Jonas Jelten*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: ge34weg@mytum.de, jelten@net.in.tum.de

***Abstract*—In this paper the main topic is the Matrix cryptography. Matrix is a system that helps humans and machines to communicate over different ways. Matrix tries to be the main platform to communicate. Many existing platforms are missing the save encryption. This subject affects most people at the moment, because the issue of safe communication is a very important topic in view of the fact that many people use these services. Matrix provides safety with the two encryption algorithms Olm, for conversation between two, and Megolm, for group conversation. The paper concentrates on how these algorithms work.**

***Index Terms*—matrix, olm, megolm, encryption, double-ratchet algorithm, communication, iot, bots, webrtc, video-telephony, messaging apps**

1. Introduction

WhatsApp, Telegram, Slack, Line and many more are platforms we daily use to communicate with others. But if you want to organise something, often a problem will arise. Namely one or more of the concerned persons does not use the platform you want to use to communicate. This is because of the high fragmentation that exists in the communication business. Matrix is created to get rid of this problem. It should help you to reach all people, who are at least registered at one platform.

Another problem, which is present in our daily life, is the safety of our data. Recently data were stolen from Facebook [1] again and everyone considers, which platform should be used to guarantee that our personal information does not get stolen. Matrix provides safety, because firstly the data are not saved at one server, where many information can be stolen at once, but on many different servers. Also the severs save only the encrypted versions of the messages, the addressee and the sender are the only ones who can decrypt them.

As you can see these issues affect everyone who does not want to switch between platforms all the time. Also it is easy for developers to integrate existing platforms to the big network. Matrix tries to connect many platforms, to make modern communication easier, like we know it from using Email.

The main topic this paper handles is how the safety of the data is guaranteed. Mainly two algorithms are used, Olm and Megolm. It is described detailed how they work and given a little overview about the features Matrix provides.

The paper is grouped in three main parts. The first handles the topic, what Matrix is and what it can be used for. The second section describes the Olm algorithm, details the double-ratchet algorithm, which is the origin of Olm, the initialisation, the main algorithm and the difference of the two described methods. The last part handles Megolm, also with the initialisation and the main algorithm in detail.

Figure 1: The Matrix Logo [2]

2. Related work

The next section is about the work that is related to this paper. First we should remind you that the topic is a very new one. The developing of Matrix started in 2014 and there are a few to none papers about this topic.

But Matrix is not the first attempt to standardise online communication, there were others. But all failed and from the mistakes that were made the developers of Matrix tried to learn.

Because the topic is very new, mostly literature is used, that is provided by Matrix itself. Matrix is a open source project, so many information about the detailed encryption process is available. Even the code itself is public.

3. Matrix

At the following part we describe what Matrix is and what it is used for. Matrix is a decentralised communication network. Decentralised means it has no main server on which everything is saved, but the data is duplicated at every participating server. It supports encrypted one-to-one communication and also group messaging. Matrix also provides real-time synchronisation and the messages that are send in JSON format are saved on all participating servers. [2]

The one-to-one communication encryption is based on the double ratchet algorithm and is called Olm, whereas the group communication is encrypted with Megolm. Olm and Megolm will be described and explained later in this

doi: 10.2313/NET-2019-10-1_05

paper. These algorithms guarantee end-to-end encryption, which means that the messages are saved encrypted at the servers and only the addressee can decrypt them again. [2]

The development of Matrix started in 2014 with a team, who was employed by Amdocs to work on this project, and since 2017 they founded their own independent company, which is called New Vector. The main team, consisting of circa twelve people, is supported by many other developers. [3]

The servers are saving the history of the communication. When a client sends a message, it will first be added to the path at his own server and then sent to the other servers. There the message will be checked, whether the sending client is really him and if the client can transmit messages. If everything is correct, the message will be added to the server's history. It can happen that two or more clients send their messages at the same moment, then the history graph splits and when the concurrent situation ends, the paths are merged together again, like it is done in Git. Because of this handling the histories of the servers are always the same. [2]

3.1. Usage

Now we describe the many different ways the Matrix network can be used. It can be used to connect messaging apps like Telegram, WhatsApp and Slack. The network supports interoperable communication, so that not everyone needs the same application to chat. The usage can be compared to Email, because writing and receiving them is not connected to the program you use. Building new bridges from existing messaging applications to Matrix is easy. For example the link to Slack has fewer than 100 lines of code. Matrix can also be added as a chatroom to other Apps, who do not use any chats until now. Encryption, Emojis, file transfer and many more features are possible with Matrix. [2]

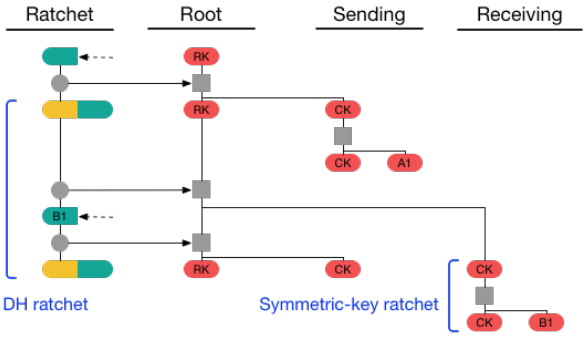
Matrix can be used in the Internet of Things (IoT), which is for example used in cars and drones. Matrix can connect different IoT silos and support them to communicate. The information gained from the silos can be published directly from the device under the user's control. Until now the fragmentation in the IoT is very high and Matrix can help to solve this problem. When developing a new device, the developers are also able to directly work with Matrix. [2]

Another usage is for Voice over IP and WebRTC, so phoning and video-telephony and many other things are possible. So far there is no standard protocol for this kind of communication. Matrix tries to become that, because it is build simple and familiar for Web developers, so they can integrate it easily to their Websites. It can be used in Apps too. [2]

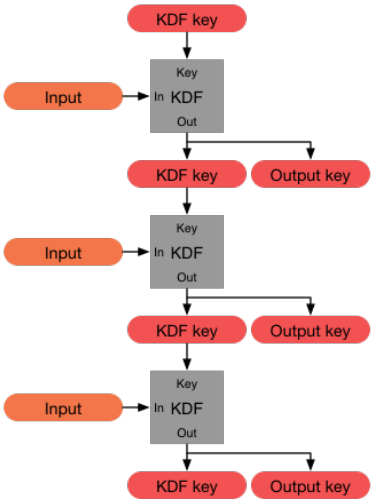
The last described way to use it, are bots. Bots must be developed for every platform separately, but with Matrix they only need to be programmed for one. [2]

3.2. Features

Now additional features of Matrix are shown that can be used, in a room. In a chatroom you can see, if the other users are online, typing or if they have already read your message. Also you can adjust how often the server informs you that new messages are available. The server can be searched to find old messages. Additionally the account data of every participant in a room is saved. [2]



(a) Chain at the double-ratchet algorithm [4]



(b) The KDF key chains [4]

Figure 2: Double-ratchet algorithm

4. Olm

At the following part we describe the Olm algorithm, which is based on the double-ratchet algorithm. Olm is used to guarantee end-to-end encryption in 1:1 communication.

4.1. Double-ratchet algorithm

At this section we look at some parts of the double-ratchet algorithm, which is the base of the Olm algorithm. It helps to understand the explanation of the Olm algorithm, which follows later in the paper.

Every message is encrypted with its own key, so hacking the system is harder. These keys are generated with KDF chains. These chains take a secret and random KDF key and input data and produce output data, which is then split in an output key and a new KDF key for

doi: 10.2313/NET-2019-10-1_05

TABLE 1: Olm Pre-key Message Tags

Name	Tag	Type
One-Time-Key	0x0A	String
Base-Key	0x12	String
Identity-Key	0x1A	String
Message	0x22	String

TABLE 2: Olm Normal Message Tags

Name	Tag	Type
Ratchet-Key	0x0A	String
Chain-Index	0x10	Integer
Cipher-Text	0x22	String

the next step in the chain with new input data. All clients have three chains, one for sending, one for receiving and a root chain. [4]

One part of the double-ratchet algorithm is the Diffie-Hellman ratchet. Every client has a Diffie-Hellman key pair with a public and private key. The sender of a message sends the public part at the beginning of the message and if the addressee does not know this key, he creates himself a new key pair. When a new key is generated, another output is also created. This is called a Diffie-Hellman ratchet step. The result of this algorithm is a constantly changing key pair. [4]

The output of the Diffie-Hellman algorithm is used to produce new sending and receiving chains, because it works as input for the root chain. The output from the root chain is then used as a new KDF key either for a new sending or a new receiving chain. [4]

The output data from these chains are used as message keys. The message keys from the sending chain are used to encrypt the message and the keys from the receiving chain to decrypt them. The inputs in these chains are constants. This is called the symmetric-key ratchet and it works because both participants start with the same Diffie-Hellman key pair and so all the following chains have the same outcomes. The only difference is that the sending and receiving chains are switched. [4]

4.2. Initialisation

Now it is shown how a room must be initialised to make Olm possible. At first one participant publishes the public part of his identity key and some single-use keys. The other participant takes the identity and one single-use key and builds his own single-use key. With the identity keys and the used single-use keys a shared secret is made using the Diffie-Hellman algorithm. This shared secret is then used to generate the first root key, the first chain key and a ratchet key. [5]

The next step is that the second participant sends a pre-key message to the first. Pre-key messages consist of a version byte, which is usually 'x\03' and payload bytes. The payload bytes have key-value format, in which the keys are encoded. The last three bits of every encoded key will show, if the following value is an integer or a string. Encoded strings have first a specific tag followed from his encoded length and then the string itself. Integers also have a tag followed by a byte, which saves the least significant bits from every Byte of the original integer. These tags for the different values are demonstrated in table 1. After that the other bytes are stored, with the high bit switching between one and zero followed by the remaining seven bits. [5]

To send a pre-key messages a new chain key and with that a new messages key is derived from the old chain key. The message contains: the public part of the senders identity key, of the ratchet key and of both single-use keys, also the current chain index and of course the message, which is encoded with the message key. The sender keeps sending these pre-key messages, till the addressee responds. [5]

When a participant receives a pre-key message he builds his root and chain key from the identity and single-use keys. The current state of the chain key can be replicated because of the received chain index. With that information he also is able to get the message key and decrypt the received message. [5]

4.3. Main Algorithm

From now on we describe how the algorithm works after the initialisation. At the beginning it is important to know, that the chain keys with an even number are used to encrypt messages from the first participant and the odd ones are used for the second participant. To send a message the sender will check, if a fitting chain key exists, or else he will create a new ratchet key. With that ratchet key a new chain and root key are generated. With the current chain key a message key is build and the message is send. [5]

A normal message consists of a Version Byte, Payload Bytes and Message Authentification Code (MAC) Bytes. The Version Byte is 'x\03' and the Payload Bytes are encoded like the pre-key messages. For the normal messages other tags are valid. These are shown in table 2. The information carried in these Bytes is: the chain index, to find the fitting message key, the public ratchet key and the encrypted real message. The MAC Bytes are part of the MAC, the length is provided by the encryption algorithm. [5]

When receiving a message, the addressee first checks if the ratchet key he receives is the same as his. If not he computes the next ratchet key and with that a new chain and root key. Also he checks if the chain indexes are the same, then he builds a new message key from the chain key, else he takes an old message key, that fits the index and was saved before. With that message key he can now decrypt the received message. [5]

4.4. Differences between the Double-Ratchet Algorithm and Olm

The biggest difference between the Olm and the double ratchet algorithm is that Olm has no sending and receiving chain, but just one in which the index decides about the sender. The Diffie-Hellman key is called ratchet byte in the Olm algorithm.

5. Megolm

In the following part we describe the Megolm protocol, which can protect the communication of many

doi: 10.2313/NET-2019-10-1_05

TABLE 3: Megolm Message Tags

Name	Tag	Type
Message Index	0x08	Integer
Cipher-Text	0x12	String

recipients in a conversation. Every member of the group has an outbound session, with a ratchet chain and a key pair. The key pair is used to authenticate him, so everyone in the group knows, who is sending and who's receiving the message. With the ratchet chain, new message keys are generated, so the safety is guaranteed. If a member wants to share his current ratchet key and his public key, he does this with a peer-to-peer connection to another member. This connection is encrypted with a safe algorithm. For example Olm can be used. [6]

For a safe storage of the server history, like it is provided by Matrix, the servers only save the encrypted messages. The users can only read these from the point when they joined the group, because all used message keys can be built from the first ratchet key they got. [6]

5.1. Initialisation

Every session of each member of a group has a counter, a key pair and a ratchet with four different values. There can be many session in a conversation. The public key helps to authenticate the different sessions. To initialise such a session, the counter is set null and a random value is assigned to the other values. To add new users to this session the session data is shared over a safe peer-to-peer communication, which can be Olm. [6]

The format to share that information consists of exactly 229 Bytes. At the beginning stands the Version Byte with the value 'x\02' followed by the four different ratchet 32-Bit Integers and the public key. It ends with a 64-Bit Signature, showing who sent the data. The receiver of this data checks the signature and saves the other values. [6]

5.2. Main Algorithm

The message key in Megolm is derived from the ratchet. The number of steps that were performed on the ratchet, plus the encrypted message, is sent to the other servers. These messages have a certain format, that has a very similar format to the message from Olm. First the Version Byte 'x\03' is sent, then the encrypted Payload bytes and the MAC Bytes like in Olm. The tags for the Megolm algorithm are described in table 3. The only difference is the signature Byte that is sent at the end, to authenticate the sender of this message. Because the messages are encrypted this good, they can be sent over insecure channels. [6]

Every message should be encrypted with another message key, so after sending, a new key is created. To do that four different hash functions are needed. The ratchet algorithm takes the four different values and changes them after a certain number of iterations. The message key is built from a hash of the combination of the four values. [6]

The value of the ratchet and the counter are stored in the session. The earliest value of the ratchet can be saved to guarantee backward compatibility. [6]

6. Conclusion and future work

Finally it can seen that the Matrix cryptography is a good way to protect our messages. The algorithms are already used by Riot and WeeChat [2]. And also common applications like WhatsApp are using techniques like the end-to-end encryption, which is also provided by Matrix.

Also the idea of connecting all communication platforms, can help many people and make communication less complicated. Matrix makes a standard and save communication possible. To help us protecting our messages more people should use the system. A commercial for developers or for everyone could help to make this network common. At first the development should be finished, to avoid mistakes, which are not identified till yet and could be a huge security lack.

But not everything is perfect, in the Megolm algorithm were found some lacks in the protection of messages. The developing team is already working to fix them [6]

References

[1] "Facebook-Hacker klauten hochsensible Daten," https://www.welt.de/wirtschaft/article182033314/Facebook-Hacker-klauten-sehr-private-Daten-von-Millionen-Nutzern.html.

[2] https://matrix.org/blog/home/, [Online; accessed 07-April-2019].

[3] "Frequently Asked Questions," https://matrix.org/docs/guides/faq, [Online; accessed 07-April-2019].

[4] T. Perrin and M. Marlinspike, "The Double Ratchet Algorithm," https://signal.org/docs/specifications/doubleratchet/, [Online; accessed 07-April-2019].

[5] "Olm: A Cryptographic Ratchet," https://git.matrix.org/git/olm/about/docs/olm.rst, [Online; accessed 07-April-2019].

[6] "Megolm group ratchet," https://git.matrix.org/git/olm/about/docs/megolm.rst, [Online; accessed 07-April-2019].

doi: 10.2313/NET-2019-10-1_05