



Smoke Crypto Chat

## McEliece-Messaging: Smoke Crypto Chat – The first mobile McEliece messenger published as a stable prototype worldwide

21.8.23 10:21 von Gastartikel Lesezeit: 14 Min.

Mobile messengers are increasingly encrypted and it is imperative that they become more secure. We introduce the app Smoke Crypto Chat.

INHALT

Summary

Smoke Crypto Chat Messenger: Fiasco Forwarding in Cryptography

Own phone number upload not required for Smoke Crypto Chat

Emoticons & pictures can be exchanged

Further App perspectives with volunteering developers from the community

Smoke Crypto Chat is open source

The free app Smoke Crypto Chat introduced. A guest contribution by Claudia Rahmschmid and David Adams, thanks! Btw.: You can find the german article [here](#).

## Summary

Mobile messengers are increasingly encrypted and have to become more secure. And they are currently strong involved in discussions about secure end-to-end encryption on the agenda of trying out – precisely because of the end of the product life cycles of the unsafe algorithms RSA and elliptical curves like ECDSA. While new algorithms are sought that are safe against the quantum computers, the McEliece algorithm has been established for around 40 years to be secure and has now also been implemented in a messenger. And: Own, open-source, networkable (federate-able) and decentralized chat servers are seen as a perspective despite central chat services. A particularly secure messenger has now been published as an open-source project model with a stable current release: The Smoke Crypto Messenger (to be found at [F-Droid.org](http://F-Droid.org)) is the first mobile messenger to use the quantum-computing secure McEliece-algorithm worldwide and thus herald a new age in the „Third Epoch of Cryptography“ (Tenzer) and their quantum computers. A stable prototype: its strengths and development perspectives are discussed in detail below. It can be found at [F-Droid.org](http://F-Droid.org) and [Github](#)) is a first step and a pioneer in applied Cryptography.

Because of an above-average, high interest and server access, especially from the edu- and university-area, as well because of several inquiries from our readers, the article on MecEliece-Messaging, that was also published in German language, has now been translated into English.

Because, since the American Institute [NIST](#) has announced already (and [final](#)) in 2016, that the RSA algorithm depending on the key size has to be considered (more or less) than broken (also with regard to the future computing power of the supercomputers), it is important to secure the possible attacks on encryption in the upcoming age of quantum computing. The same-named algorithm of the recently deceased [Robert McEliece](#) is considered particularly secure alongside the algorithm [NTRU](#).

## Smoke Crypto Chat Messenger at [f-droid.org](#)




Now a first mobile messenger has implemented this algorithm and been released with another stable release: [Smoke Crypto Chat Messenger](#). This is stored at [Github](#) and has also undergone the strict requirements for the source code during an compilation by the alternative download portal [F-Droid.org](#).


This messenger integrates into a chat server architecture, which can also be decentralized and is federate-able (network-able). The associated server is: [SmokeStack-Server](#) and is based on the [Echo-Protocol](#), which secures the encrypted message again by a HTTPS/TLS-connection (thus means using [multi-encryption](#)).

**Not to forget, the server question is very crucial for messaging:  
Smoke communication through your own server for family communication? – The server setup on Android becomes a topic easy to learn**


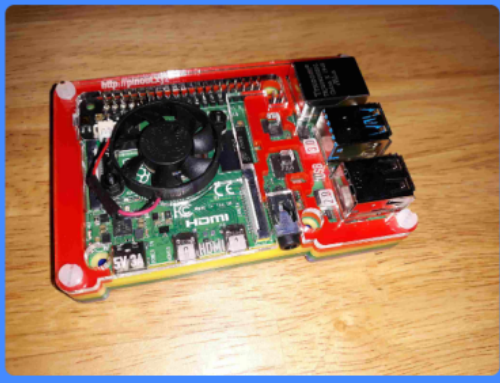
The server question for messaging communication is a very crucial one: So far, [XMPP](#) aka Jabber, has played the role of the leading decentralized server architecture, since XMPP is well documented and also used frequently. However, their servers are not all with the [XEP](#)-extensions for encryption technically compatible and also not encrypted in the implementation or do not pass the compliance-tests. In addition, even servers that can implement encryption such as [Ejabber](#) or [Prosody](#), are often only installable for experts. Alternatives such as [Matrix](#)– oder [Signal-servers](#) are even for IT-interested people hardly installable, not to speak of compile-able or open source – or you need an Amazon AWS-Docker access or have a TOR-Node-DHT on your device. And and and..


The [SmokeStack-Server](#) (also at [F-Droid](#)) is not just open to source here, but much easier to administer: it is an easy-to-use application that runs under Android. Smokestack thus describes simple mobile communication servers for your pocket in the jeans, your own home or for the classroom. Other Echo-Servers (such as [Spot-On Chat Server](#) oder [Spot-On-Lite](#)) exist for numerous operating systems und bigger scales.


8:14   100% 



**Smoke | nausicaa@D4DE-2515-66BE...** 


Connected (192.168.178.220:4710)


   
2021-02-28 8:13:55 PM


 Greetings.  
2021-02-28 8:11:44 PM

 Yo.  
2021-02-28 8:11:31 PM

 Received a half-and-half call.  
Dispatching a response. Please  
be patient.  
2021-02-28 8:11:10 PM 

 Greetings.  
Ozone 2021-02-28 8:09:03 PM

 Hi.  
2021-02-28 10:27:44 AM

 Hi.  
2021-02-28 9:50:26 AM




 Please type a message...  

Figure: Current Graphical User Interface of the 1:1-Chats in Smoke Messenger under Android.

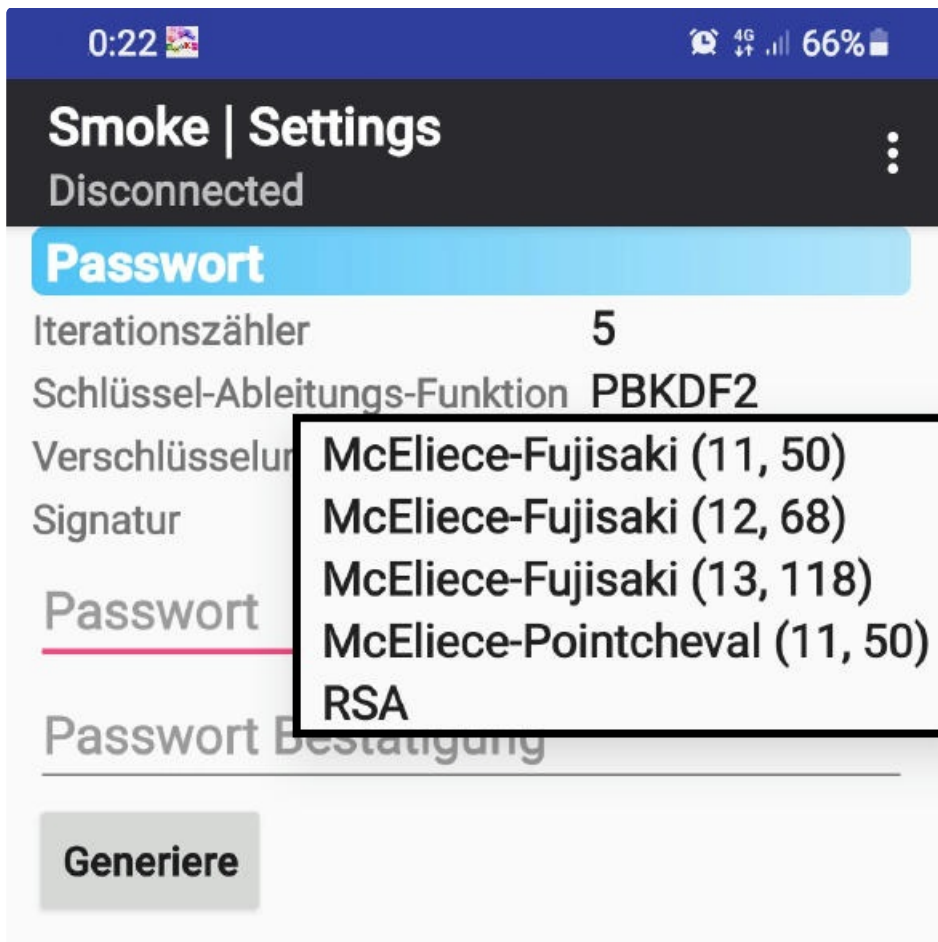
IT-teachers at every school can set up and administer an open-source communication server for their class without further dependencies. And, that for that purpose there are various clients and customized user interfaces available – at best, which look like popular messengers like WhatsApp. An encryption handling chat-server should be able to be set up by any IT teacher at school.

With this type of Echo-server, with the Smokestack application, the server setup is very easy to use via an Android app – the actual messaging app Smoke enables with Smokestack to put on an own server and therefore opens up all interested persons access to quantum-computing more secure McEliece-Messaging.

## **Smoke Crypto Chat Messenger: Fiasco Forwarding in Cryptography**

The Smoke Chat client is not only open source and equipped with the McEliece encryption, it also enables the user to get defined passphrases in the sense of „Customer Supplied Encryption Keys“ (CSEK) for a safe end-to-end encryption: the Cryptographic Calling, an end-to-end encryption in the sense of the Forward Secrecy with temporary keys can therefore be carried out with self-chosen, individual passphrase on both sides (symmetrical encryption).

Temporary asymmetrical keys with the algorithms McEliece (4 different moduli) respective RSA can also be used in the Messenger Smoke: A public key infrastructure (PKI) is used, i.e. with a private and a public McEliece key. The mathematical highlight: RSA key users can also chat with McEliece key users. The messenger is a good practice model to train IT apprentices by getting to know the McEliece algorithm and learning mathematician students how RSA keys with McEliece keys become interoperable.



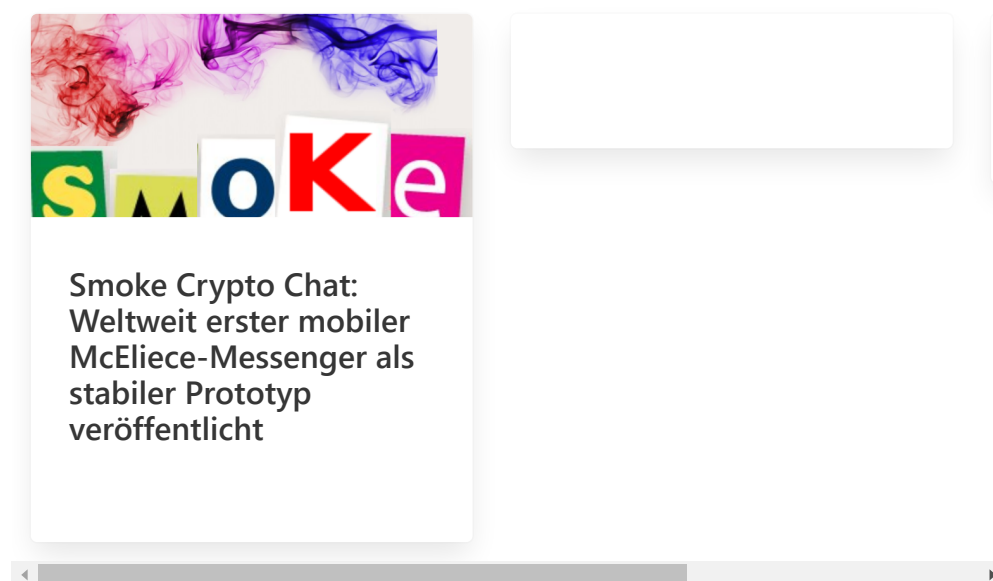
*McEliece-Messaging with four different Moduli as Choice for the Encryption-Algorithm McEliece.*

In addition, so-called **Fiasco-Keys** are implemented in the Smoke Messenger: These are also a whole bundle of temporary keys in the sense of the Forward Secrecy for encryption and are kept in a cache for the corresponding decryption attempts. XMPP with compatible mobile clients, historically, is therefore more likely to be located in development from encryption, in which only one key is sent per session (OTR encryption), which is now the **OMEMO** encryption with a key per message.

The current further development is: Fiasco Forwarding (FF) is compared to the Omemo respective also Signal Protocol, in which the (one) key of the upcoming message is derived very schematically from the key of the previous message, much more volatile, i.e. encryption becomes safer by **Fiasco Forwarding (FF)** – and messaging also becomes a safer process through this process design (also described by **Tenzer**). If a key is known in the Signal protocol (and thus also in WhatsApp encryption), follow-up messages could also be deciphered, which can then also refer to the other keys – and then open up in the stocking like a stitch. Security is there with a purely schematically assigned and permanent method derived key (so-called **Double Ratchet** procedure).

The protocol of the Fiasco-Forwarding in the Echo-Protocol is therefore more mature here, since it leaves numerous options: The signal of evaluating a development trend in Cryptography is about the further development of a key transmission protocol: from the Signal, formerly Axolotl or Omemo protocol, to Fiasco-related, volatile options in Cryptography, in which much more than one key are sent per message! Perfect-Forwarding (of one key per message) has a successor: Fiasco-Forwarding (of a bunch, up to a dozen, of keys).

Mehr zum Thema



## Own phone number upload not required for Smoke Crypto Chat

The chat-ID is regulated in the Smoke Crypto Chat via a 32-digit number, so that the user does not have to announce an own telephone number. The automatic telephone book upload is particularly the illegal non-conformity of other messengers with risk in Law infestation for some countries following the [GDPR](#). Over this ID, based on the [SIP-Hash](#) process, used for a symmetric encrypted channel, the cryptographic keys are safely transferred.

Since no one can notice (or wants to type in) such a long ID-number, there is the further option of using an Alias. A chat user defines an Alias term in the chat client, for example the city name „Boston“, since he lives there, and the chat friend also gives his Alias to this user. Complete. The McEliece key is then exchanged via this „Boston“ channel and the secured chat can begin (then via McEliece).





This means that central key servers have been replaced by the autocrypt function. With autocrypto, the applications of two chat partners automatically exchange the public keys. Autocrypt goes back to the idea of a REPLEPO, in which even a user's public key is only encrypted with the key of the friend. It is used, for example, by the promising messenger [DeltaChat](#) and others who use chat via email servers (using the [POPTASTIC](#) protocol). Now the Smoke Messenger comes and converts the path from the key server to Autocrypt one step further: Auto-Crypto is replaced by the Alias. A simple key based on the selected password (e.g., „Boston“) – which would not be incompatible with any encryption control regulation – creates a channel that then shares the safe public key. The SmokeStack-Server also has special key server capabilities.

## **Emoticons & pictures can be exchanged**

In addition to text, emoticons and pictures can also be sent in the chat. The latter are stored in an encrypted container on the smartphone. This means that an image must also be extracted from it before it can be continued unencrypted. Smoke is known as one of the only messenger, which has a login and also encrypts the saved databases as a safe container.

Smoke Crypto Chat is one of the few messengers that not only require a security password to log in, but also to decrypt the data on the smartphone's Android hard-drive. It should be noted that some views and menu settings only become visible if several friends are inserted as users in the messenger. It is an exploratory model, also in the graphical interface: a pioneer model, ideal for learners.

8:15   100%

## Smoke | Settings

Connected (192.168.178.220:4710)

### About

Bouncy Castle Version 1.68  
 Smoke Version 2021.02.27 Solid Smoke (Final) (Debug)  
 Build Date 2021-02-28 10:31:24 UTC  
 Android 9  
 WakeLock Locked: False  
 WiFiLock Locked: True


Foreground Service  
 Prefer Active CPU  
 Query Time Server

Clear Log

### Neighbor Servers

Control	Remote
Action ▼	Control: Connect Status: Connected 192.168.178.220:4710:TCP 192.168.178.195:59009 Passthrough: false Proxy: Temp. Queued: 0 / 256 In: 2.25 MiB Out: 52.15 KiB Outbound Queued: 0 Uptime: 4:01 Min.

Automatic Refresh  Details

Echo 

Refresh

There is a separate function for the file transfer of any files (in any size) from cell-phone to cell-phone. The special thing here: the files can also be sent to a SSH client and do not need a second Smoke client. This standard also relies on interoperability using the [Steam-protocol](#).

Messages to offline users are temporarily saved in a so-called [Ozone-Postbox](#), whose name only in the client Smoke and Server SmokeStack must be set equally; The cryptographic keys in the Smokestack-server automatically regulate everything else. The group chat is currently established in IRC style based on symmetrical encryption and compatible with the client of the [Encryption Suite Spot-On](#), which, in addition to the server function, also includes numerous open source encryption tools.

The SmokeStack-Server is also a key server for the user (which XMPP-servers and others are not, for example). Even from the Smoke client, a friend's public key can be forwarded to the direct IP-connection for faster networking to servers or friends.

## **Further App perspectives with volunteering developers from the community**

The Messenger-Client Smoke has been the world's first mobile McEliece Messenger since 2016 (in addition to further implementations on the desktop). It is fully functional with the current release. The easy to set up server enables everyone to get to know the operation of the new elements. A further adapted and polished user interface (especially for the individual settings) could simplify the operation here. An Italian development team has already started a new development of the user interface for the iPhone. They „forked“ the project for its own development in swift respective flutter programming language.

However, wishes to make Smoke Crypto Chat look like [WhatsApp](#) are disappointed and remain in this project model development tasks, which, however, are not all Rocket Science. It needs organizations such as universities and highschools that still develop instead of going to buy on the market. Or free and interested GUI & Java developers for Android from the common good-oriented community environment, which create a chat client based on the simple server Smokestack with simple architecture of the HTTPS shipping of a with-McEliece-encrypted message.

If this prototypical model concept were further adapted, for example, as a separate development and chat client, this chat app could then use any school without much administration effort. The future is neither in the proprietary protocol, the libraries for the code, nor in the niceness of one App. But in the simplicity of a chat server to be set up in a decentralized chat in every classroom of an IT-teacher and the (financial savings) goal also for organizations such as communities and municipalities, to have developed their own open-source messenger for the members and to be able to operate themselves inexpensively.

After all, it is also about state and municipal independence from central communication servers in the architecture of the software landscape and the data protection-compliant design. And this without uploading phone numbers of friends.

If one wants to become (technical) and (also) decryption **world champion** one has also to accept the leadership in the development of open source cryptographic messengers and take over their learning content and adopt their teaching and learning content – and „**dare more encryption**“. At the moment, a lot of time seems to be spent on elaborated discussions in terms of privacy and encryption of private electronic messages from citizens beyond the technical possibilities.

## **Smoke Crypto Chat is open source**

The prototypes Smoke and SmokeStack with their code base are not only suggested for evaluation, but also for everyone discoverable. Open source enables students to learn together in the field of cryptography, e.g. using this pioneering McEliece messenger. The future belongs to **research** into McEliece messaging with an algorithm that the security institute **BSI** in Germany and also probably following the American Institute for Standards and Technology **NIST** are currently researching for a future standardization of encryption in the epoch of quantum computers. To the general and interested public this messaging technology is already available open source today.