Homework #5
Telegram vs WhatsApp Security

CNS Course Sapienza

Riccardo Prinzivalle, 1904064

November 30, 2020

1 Homework Goal

This homework contains a comparison of Telegram and WhatsApp security, with an in deep to protocols comparisons and past and some current threat an vulnerabilities.

2 Telegram Security basics

Telegram uses a security protocol called MTProto, developed by the telegram team. It is a symmetric encryption protocol based on 256-bit symmetric AES encryption, 2048-bit RSA encryption and Diffie–Hellman key exchange. The protocol is divided in 3 layers:

- **High-level** component which defines the method whereby API queries and responses are converted to binary messages.
- **Cryptographic/authorization** layer which defines the method used to encrypt messages prior to being transmitted through the transport protocol.
- **Transport** component, which defines the method for the client and the server to transmit messages over some other existing network protocol.

Let's analyze in the details every section. The high level component sees a client and a server exchanging messages inside a session, which is identified by a user key identifier (a particularity, the session is attached to the client instead of standard protocols such as http/s or tcp). The client can instantiate different connections to the server (the practicality of Telegram stands in the fact that one can open different sessions on many devices such as browsers without having to log in many times once one have the session active), and messages can be sent from one connection to the other and everything is synchronized server side. The low level message structure can be seen in fig. 1.

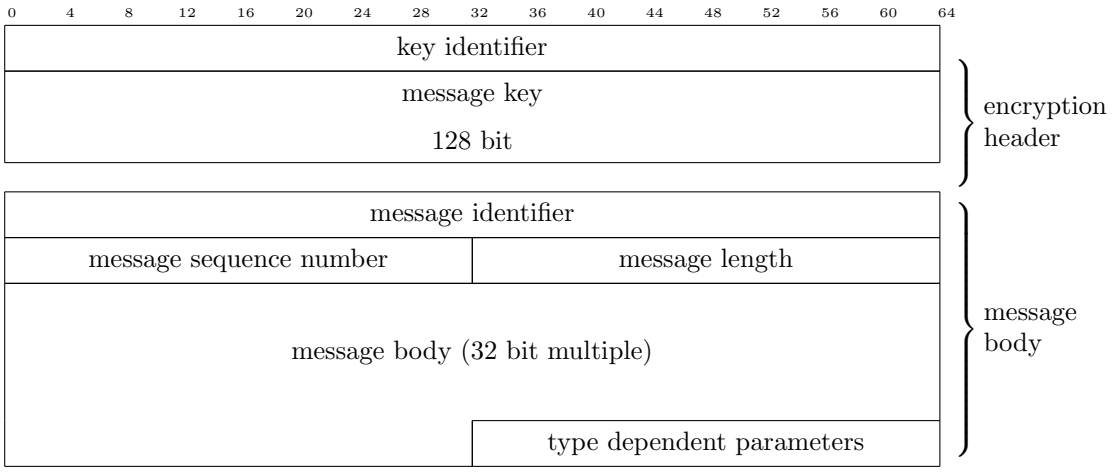

Figure 1: Telegram simplified packet structure

To be mentioned, all number are saved as little endian, with the exception of large numbers, such as those needed by RSA and DH which are stored as big endian due to openSSL compatibility.
The encryption part can be identified in the upper part fig. 1: the message is encrypted using a 256-bit key constituted by the message key and the user key and the encryption is performed with AES-256. The message key is defined as the hash using SHA256 on the message body, and taking the 128 middle resulting bits. The user key is generated from the authorization key: it is created once, when the client is first run on the device, and never changes, so this will expose all messages if that key is stolen (even from the device or from server side); different counter measures can be taken:

- Use session keys generated at every session using the Diffie Hellman exchange protocol
- Store the keys on the device and protect them with a password
- Protect all stored and cached data of the device with a password

All these measures cannot protect the user in the case where is the server that is violated or some government agency ask the keys for terrorism prevention (as example). The complete encryption scheme for every message can be seen in fig 3.

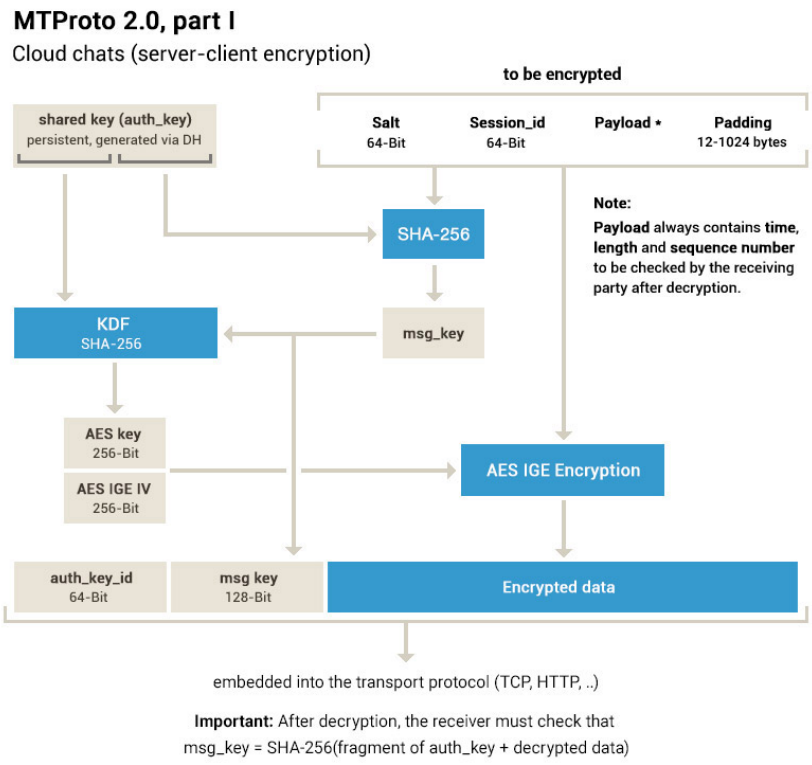

Figure 2: MTProto

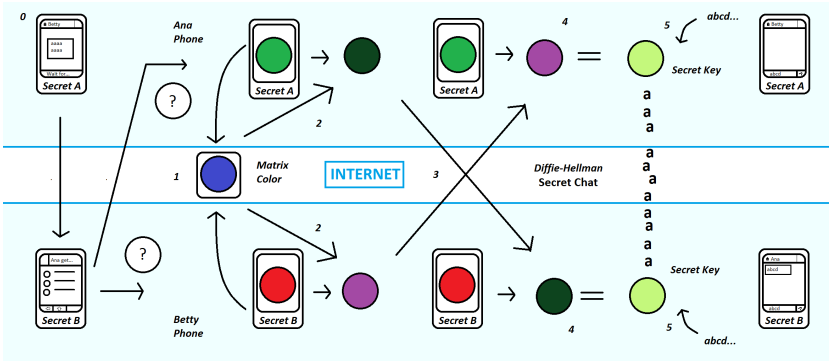

Figure 3: Telegram encryption scheme [1]

3 WhatsApp Security Basics

4 Protocols comparison

5 Past Evolution and Fixed Vulnerabilities

6 Current Vulnerabilities and Security Threats

7 Conclusion

After this brief introduction on elliptic curves, it is obvious why they have been widely adopted in many cases of asymmetric encryption: they use less bits for the same level of security, so are more efficient to compute and their base concepts are easier to visualize. As suggested on section , EC can be easily used in hybrid encryption scheme, in the key exchange phase.

References

[1] D. Sanguinetti. `https://commons.wikimedia.org/w/index.php?curid=36531862`.