E2EE and Instant Messaging: Signal Protocol

Yaman Yağız Taşbağ
@yamantasbagv2

ExploitStudio Bilkent
@ExploitStudio

November 28, 2019

Table of Contents

1. Introduction

2. Key Exchange Protocols
   - Diffie Hellman KE
   - Triple DHKE
   - Extended 3DHKE

3. Message Exchange
   - Authenticated Encryption and Additional Data with Counter
   - Symmetric Key Ratchet
   - OTR Ratchet
   - Double Ratchet

Introduction

Goal: Develop a *face-to-face* equivalent.

Face-to-face:

- Authentication
- Confidentiality
- Repudiability
- Forward Secrecy
- Passive Backward Secrecy

Signal Protocol

First version in 2013. Formerly known as *TextSecure*. Used by many Instant Message(IM) applications: *Signal*, *WhatsApp*, *Facebook Messenger* and *Skype*.

Why TLS does not work

- It is build for Client-Server infrastructure.
- Both parties have to be online.
- It is non-repudiable.

Why do we want End-to-End Encryption(E2EE)

Without E2EE the model is as follows:

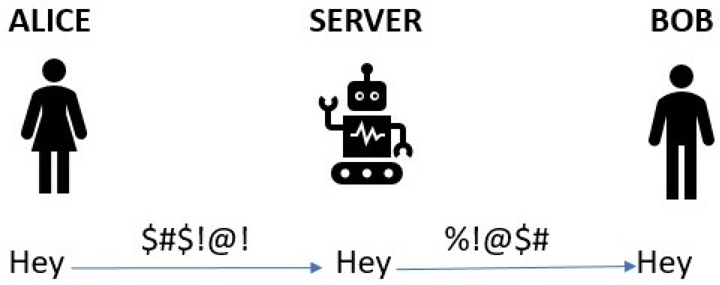


Figure: IM without E2EE

Why do we want End-to-End Encryption(E2EE)

With E2EE the model is as follows:

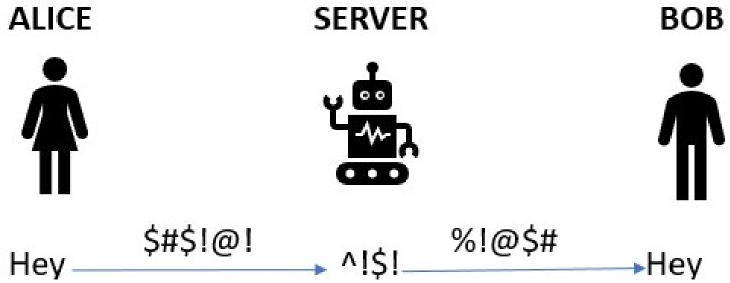
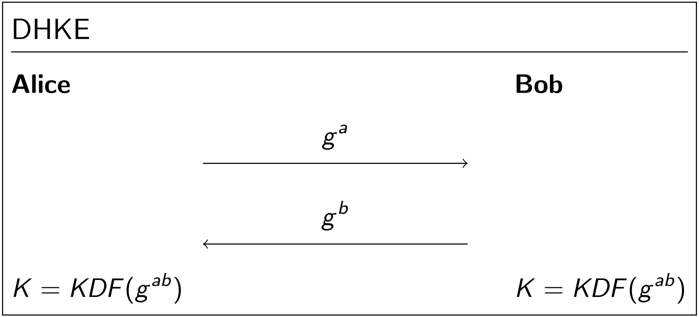


Figure: IM with E2EE

Diffie Hellman Key Exchange

Accepted as the beginning of Public Key Cryptography.



DHKE

Alice **Bob**

g^a

g^b

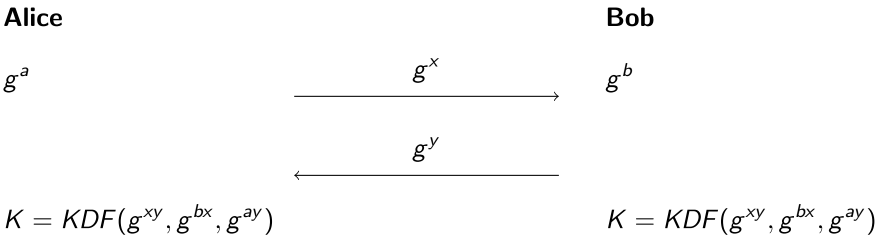
$K = KDF(g^{ab})$ $K = KDF(g^{ab})$

Diffie Hellman Key Exchange

- Authentication ✓
- Repudiability ✗
- Forward Secrecy ✗
- Passive Backward Secrecy ✗

Triple-DHKE

Triple DHKE



Alice

Bob

g^a

g^b

$$g^x \longrightarrow$$

$$\longleftarrow g^y$$

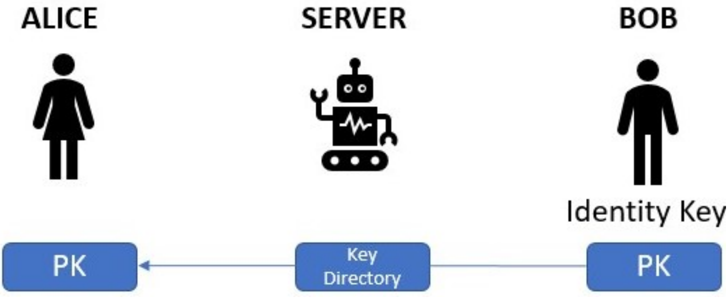
$K = KDF(g^{xy}, g^{bx}, g^{ay})$

$K = KDF(g^{xy}, g^{bx}, g^{ay})$

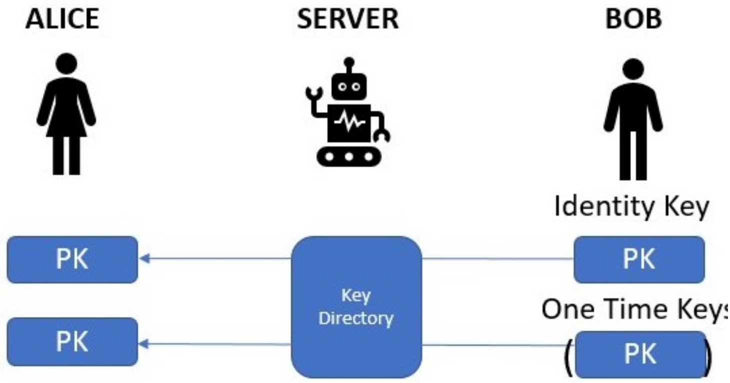
Triple DHKE

- Authentication ✓
- Repudiability ✓
- Forward Secrecy (Depends)
- Passive Backward Secrecy (Depends)

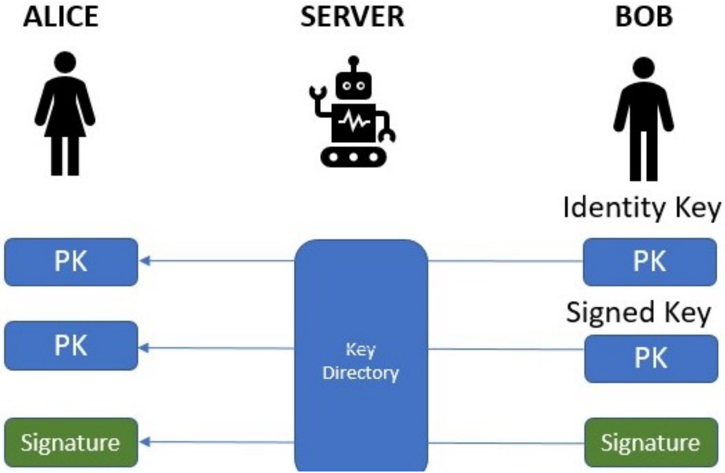
X3DHKE (1/5)



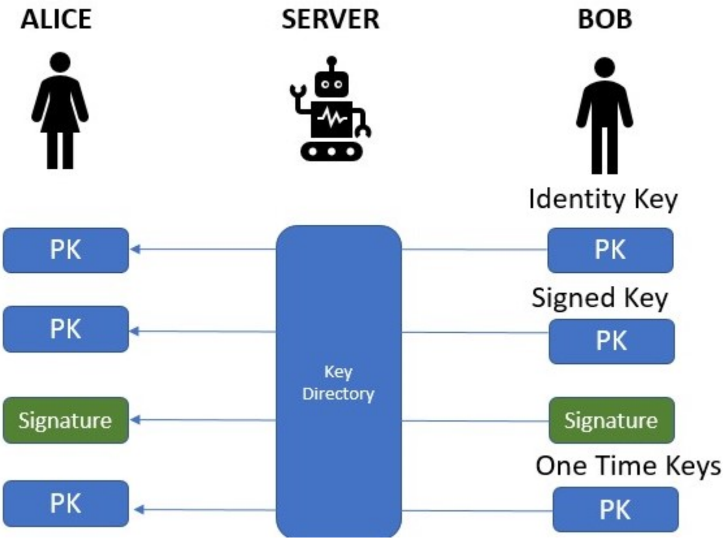
X3DHKE (2/5)


ALICE
SERVER
BOB
Identity Key
PK
PK
Key Directory
PK
One Time Keys
PK
(PK)

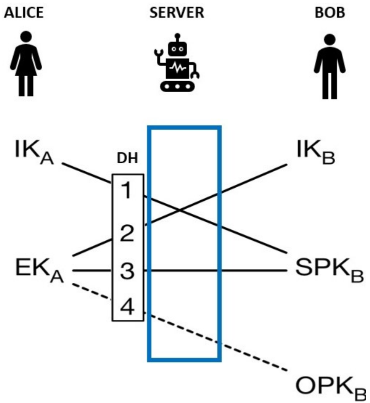
X3DHKE (3/5)


ALICE
SERVER
BOB
Identity Key
PK
PK
Signed Key
PK
PK
Key Directory
Signature
Signature

X3DHKE (4/5)

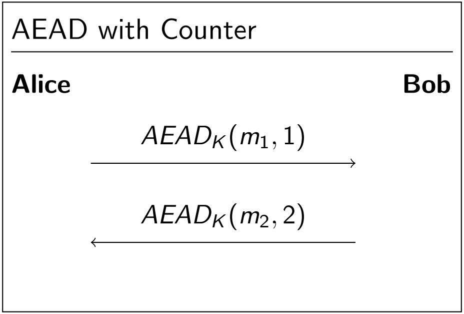

ALICE
SERVER
BOB
Identity Key
PK
PK
Signed Key
PK
PK
Key Directory
Signature
Signature
One Time Keys
PK
PK

X3DHKE



$$K = KDF(DH_1, DH_2, DH_3, DH_4)$$

AEAD with Counter



AEAD with Counter

Alice **Bob**

$$\xrightarrow{AEAD_K(m_1, 1)}$$

$$\xleftarrow{AEAD_K(m_2, 2)}$$

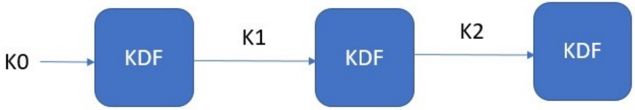
Where K is a symmetric key similar to agreed session key.
$AEAD$ provides the following:

- **Encrypted Data**: This is the message content. It is both encrypted and authentic.
- **Additional Data**: This part is the counter. It is in plaintext but it is authentic.

AEAD with Counter

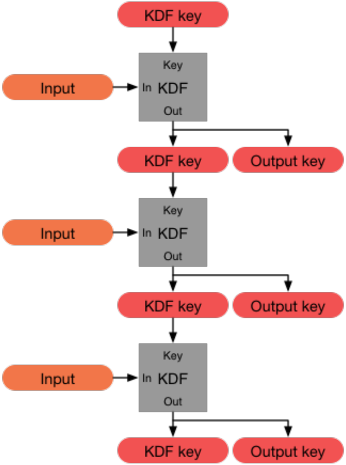
- Authentication ✓
- Repudiability ✓
  Since K is both known by Alice and Bob they cannot prove to a third party the opposite party wrote a message.
- Forward Secrecy ✗
- Passive Backward Secrecy ✗

Symmetric Key Ratchet



Uses a Key Derivation Function to rotate the keys with messages. What happens when messages appear *out of order*?

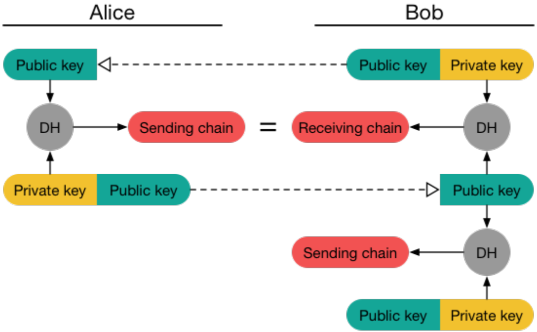
Symmetric Key Ratchet



Symmetric Key Ratchet

- Authentication ✓
- Repudiability ✓
- Forward Secrecy ✓
- Passive Backward Secrecy ✗

OTR Ratchet

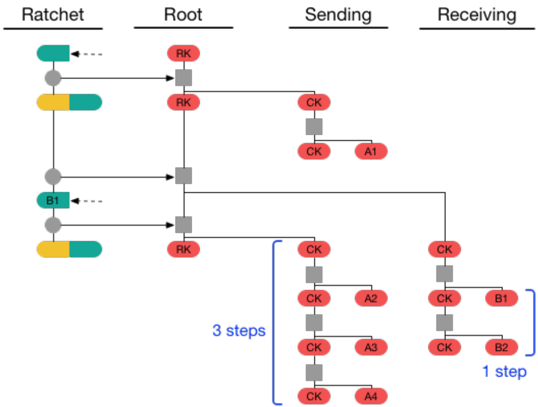


It has a *ping-pong* nature. What happens when Bob wants to share two consecutive messages?

OTR Ratchet

- Authentication ✓
- Repudiability ✓
- Forward Secrecy ✓
- Passive Backward Secrecy ✓

Double Ratchet



This way we can handle *offline consecutive* messages.

Double Ratchet

- Authentication ✓
- Repudiability ✓
- Forward Secrecy ✓
- Passive Backward Secrecy ✓

Thank you for Listening

Questions?