Ministry of Economy and Finance

End to End Encryption on Messenger

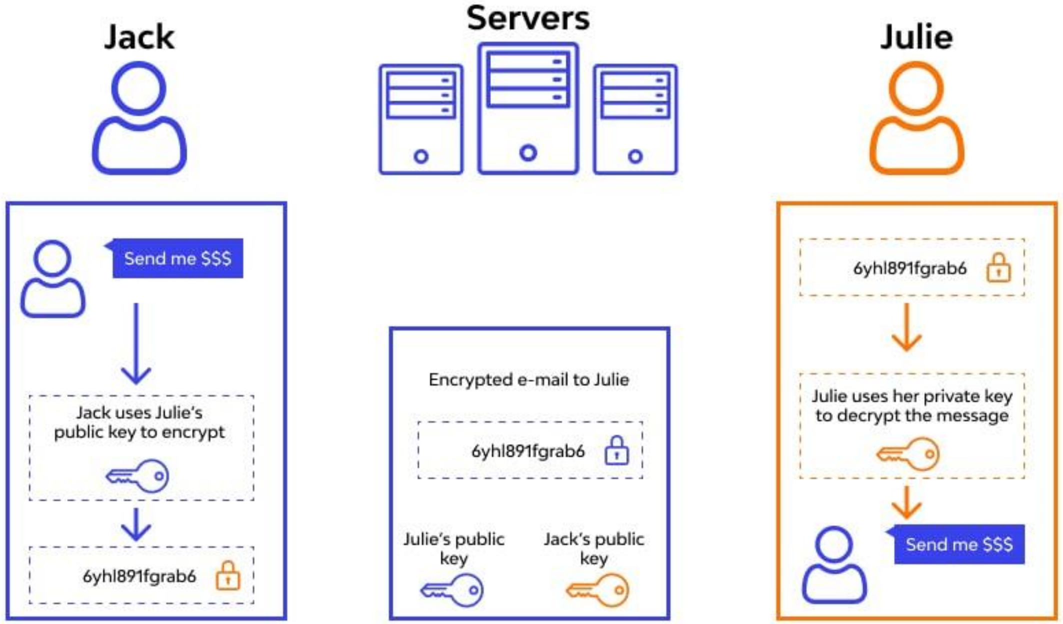
Content:

- Overview
- What is E2E encryption on messenger
- How E2E encryption protect conversations
- Protocol
- Flow Of Processing E2E encryption on messenger
- សIntegrity and Authentication
- Secure Storage

Overview:

- End-to-End Encryption (E2E) relies on strong cryptographic algorithms to secure communications.
- These algorithms ensure confidentiality, integrity, and authenticity of messages.

Overview:


Jack
Servers
Julie
Send me $$$
Jack uses Julie's public key to encrypt
6yhl891fgrab6
Encrypted e-mail to Julie
6yhl891fgrab6
Julie's public key
Jack's public key
6yhl891fgrab6
Julie uses her private key to decrypt the message
Send me $$$

What is E2E encryption on Messenger

- Messenger end-to-end encryption ensures extra security and protection for Message and calls, ensuring you and the recipient can see, hear, or read the contents. Mean that nobody else can see or listen to what sent or said not even Meta.

How end-to-end encryption protects your conversation

- Every device in an end-to-end encrypted conversation has a special key to protect the conversation. When you send a message in an end-to-end encrypted conversation, your device locks the message as it's sending. This message can only be unlocked by a device that has one of the keys for that conversation.

How end-to-end encryption protects your conversation

- You can compare **your keys** with another person's keys to make sure only your devices have access to your secure conversation. Once you're viewing your keys in your end-to-end encrypted chat, If the keys match, you know the conversation is secure between these devices.
- For example:Alyssa clicks **Your keys** on her computer and sees 123.

  Her friend Brandon clicks her name on his computer, and he sees 123.

  This means that the keys for Alyssa and Brandon match, so they know their conversation is secure on these devices.

Protocol

Signal Protocol:

Signal Protocol is a cryptographic protocol designed for securing instant messaging and voice calls. Developed by Open Whisper Systems, it is widely recognized for its security and privacy features.

Protocol Cont

Key Features:

- **End-to-End Encryption**: Ensures that messages are encrypted on the sender's device and only decrypted on the recipient's device, preventing intermediaries from accessing the content.
- **Forward Secrecy**: Uses ephemeral keys for each session, ensuring that even if one session key is compromised, previous and future sessions remain secure.
- **Authentication**: Verifies the identity of communication partners to prevent man-in-the-middle attacks.
- **Double Ratchet Algorithm**: Combines both Diffie-Hellman ratchets and symmetric-key ratchets to provide forward secrecy and post-compromise security
- **Sealed Sender**: Allows sending messages without revealing the sender's identity to the server.

Flow Of Processing E2E encryption on messenger

1. Key Generation

- **Algorithm**: Elliptic Curve Cryptography (ECC).
- **User Registration**:
  - Each user generates a public-private key pair.
  - The private key is securely stored on the user's device.
  - The public key is shared with the server for distribution.

2. Key Exchange

- **Algorithm**: Elliptic Curve Diffie-Hellman (ECDH).
- **Public Key Distribution**:
  - The server distributes the public keys of users to facilitate secure communication.
- **Initial Key Exchange**:
  - When User A wants to send a message to User B, User A's client retrieves User B's public key from the server.

Flow Of Processing E2E encryption on messenger Cont

3. Message Encryption

- **Algorithm**: Advanced Encryption Standard (AES) for message encryption; ECC for encrypting the session key.
- **Session Key Generation**:
  - A unique session key is generated for each conversation or message using a secure random number generator.
- **Encrypt Message**:
  - The message content is encrypted using the session key with AES (e.g., AES-256).
  - The session key is encrypted using the recipient’s public key with ECC.

4. Message Transmission

- **Send Encrypted Message**:
  - The encrypted message and the encrypted session key are sent to the server.
- **Server Relays Message**:
  - The server relays the encrypted message to the recipient without decrypting it.

Flow Of Processing E2E encryption on messenger cont

5. Message Decryption

- **Algorithm**: AES for decrypting the message content; RSA/ECC for decrypting the session key.
- **Receive Encrypted Message**:
  - User B's client receives the encrypted message and the encrypted session key.
- **Decrypt Session Key**:
  - User B’s client uses their private key to decrypt the session key with RSA or ECC.
- **Decrypt Message**:
  - The session key is used to decrypt the message content with AES.

Integrity and Authentication

Digital Signatures:

- Each message is signed using the sender’s private key to ensure authenticity and integrity.

Verification:

- The recipient verifies the digital signature using the sender’s public key.

Secure Storage

On Device:

- Decrypted messages are stored securely on the user’s device using local encryption mechanisms.

Server Storage:

- The server stores only the encrypted messages, ensuring that it cannot read the content.

Video (HMAC)

Thank you!☺