E2E Encrypted Chat

Team us

Till Behrendt

Jaime Park

Overview

Python client using cryptography.hazmat

Ubuntu server with Nginx. Node.js, and mongoDB

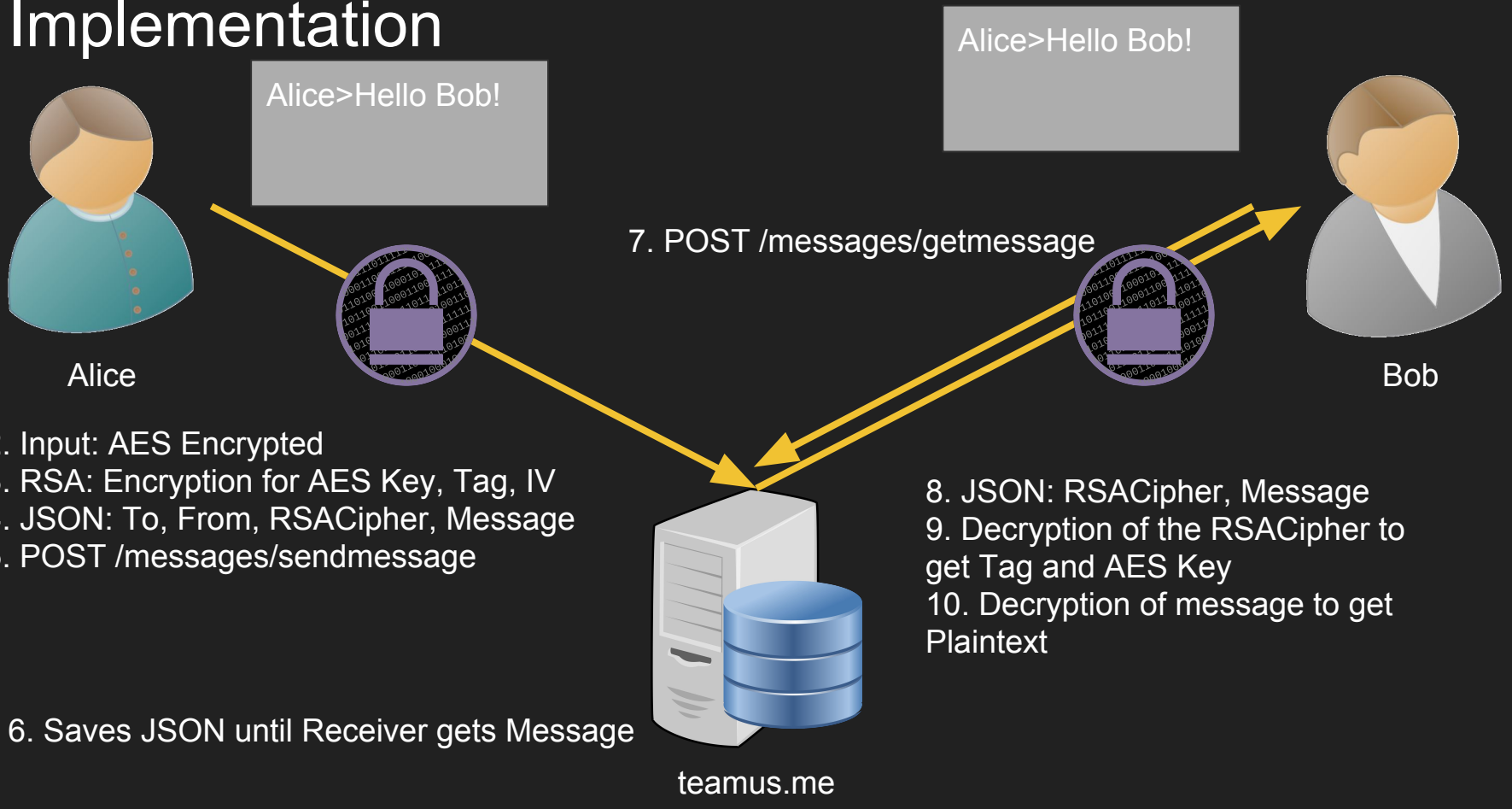
User creates an account to use with client, sign-in generates JWT, which is used to prove authenticity of messages

RSA public keys must be exchanged physically to ensure confidentiality

Clients can send messages to each other, even if one of them is offline

Even if adversary knows client’s public key, messages they send will not be received by the client because the client has not initiated a conversation with the adversary.

Implementation

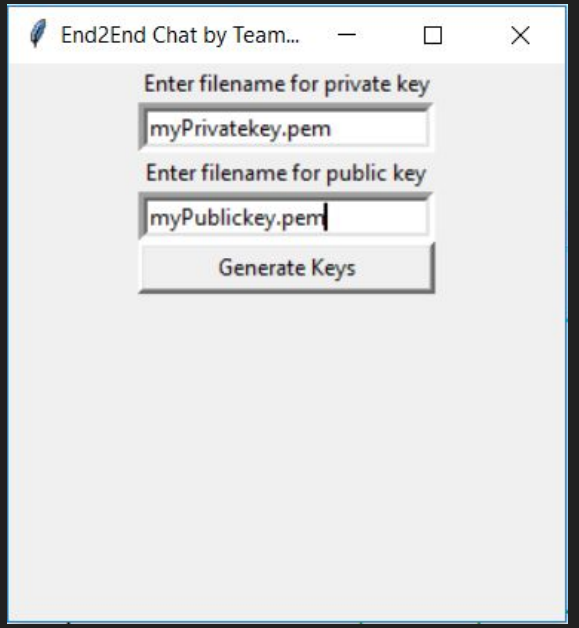


Demo

https://www.youtube.com/watch?v=CdWZ13ISCeU

Further functions

- Key Pair Generation
  - Generates a RSA Key Pair and saves it to program's directory
- User deletion
  - User has to signin first



Assumptions

- AES 256 in conjunction with RSA OAEP provides security and confidentiality over an insecure channel
- Users have securely shared RSA public keys beforehand
- Server availability is not an issue

Possible Attack surfaces

Brute force user credentials

Man in the middle attack

DDOS

Compromised server

Malicious client

Future Work

- Sign up with email, able to reset password with email confirmation
- Support for multiple clients/devices
- Group conversations
- True end to end encryption (double ratchet)
- True end to end communication (no server)
- Exchange RSA public keys more practically (QR code)

Questions?

Sources

- https://ctan.org/pkg/tikzpeople?lang=de
- http://worldartsme.com/computer-server-clipart.html#gal_post_12368_computer-server-clipart-1.jpg
- https://openclipart.org/image/2400px/svg_to_png/204439/encryption.png