SecureChat

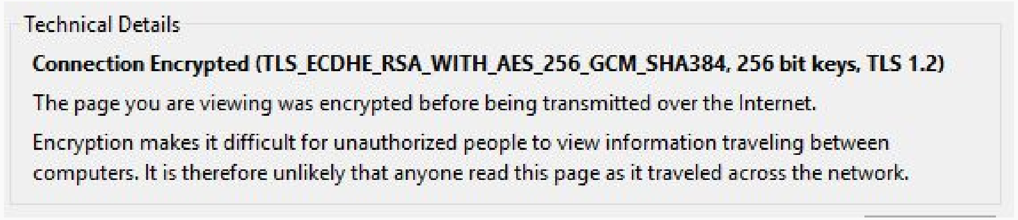
Android Application

Configuring The Server

- Amazon Web Services with Amazon Linux Server
- SSL Protocol
- TLSv 1.2
- LetsEncrypt certificates

Secure RESTful Server

- Client server relationship
  - Keep them separated
- Stateless, good!
  - All requests contain all information necessary, does not rely on any info from the server
- Layered
  - Components only have access to what they are allowed to interact with



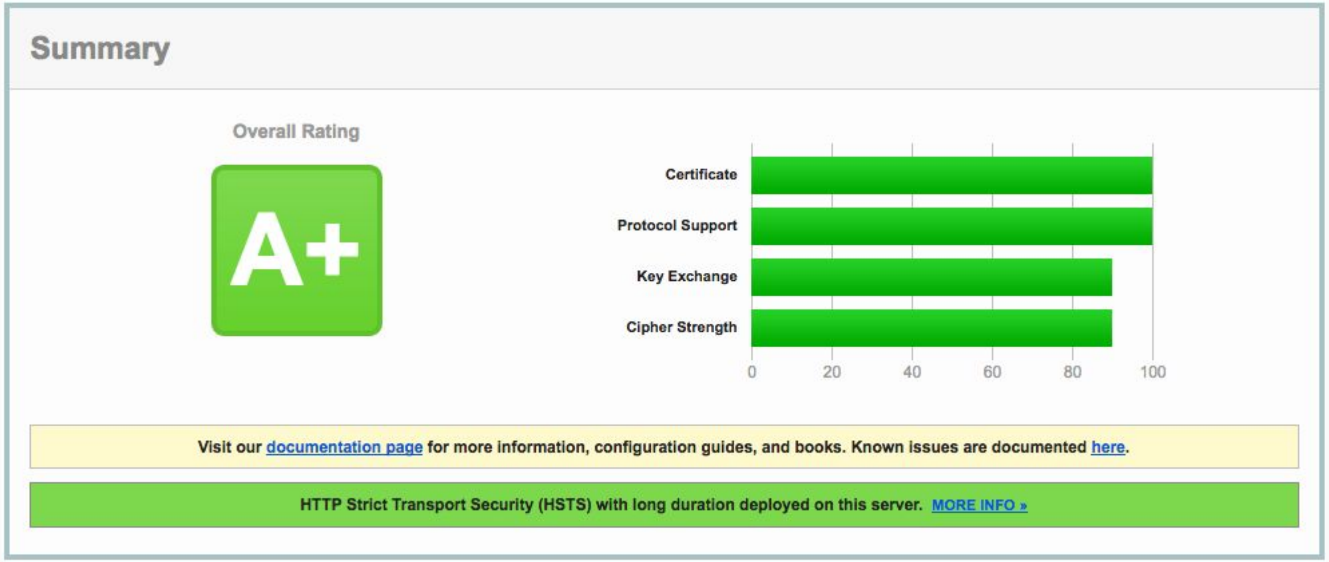
Technical Details

Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, 256 bit keys, TLS 1.2)

The page you are viewing was encrypted before being transmitted over the Internet.

Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.

OpenSSL Labs Score


Summary

Overall Rating

A+

Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. MORE INFO »

Register/Login Post

Register Post

- Username, email, password
  - Verifies uniqueness
  - Password is hashed before it is stored in the database
- Makes sure user enters required fields in the form
- Creates new user in the database

Login Post

- Username and password
  - Verifies user is in the database
- Ensures correct credentials were entered.
- Issues a JWT to the user

Bind parameters to prevent SQL injection

Send & Get Message

Send Message

- Validates that there is a message and recipient to send message
- Sends message in JSON object for client to receive.

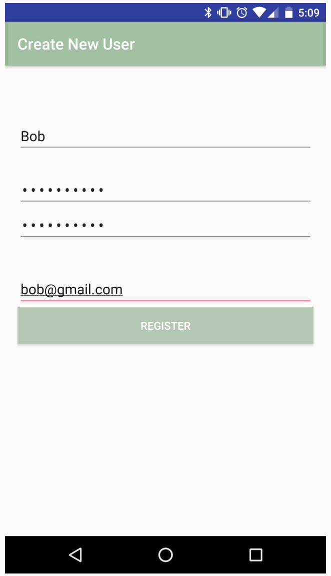
Get Message

- Retrieves messages from same sender and orders them by time received

Bind parameters to prevent SQL injection

Android Client

- **Volley** for HTTP requests
  - Make StringRequest, then convert to JSONObject
- **SpongyCastle** for cryptographic libraries
  - Same as BouncyCastle but tweaked a little to work with android
- **ZXing (Zebra Crossing)** library for generating and scanning QR codes



Bob

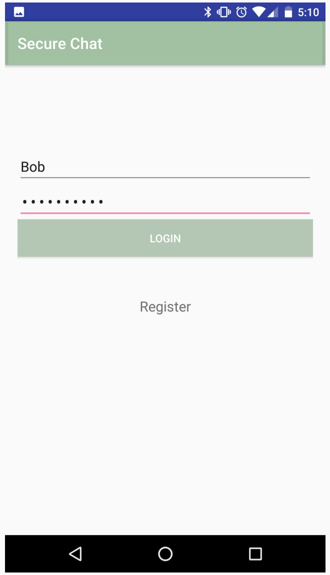
....

....

bob@gmail.com

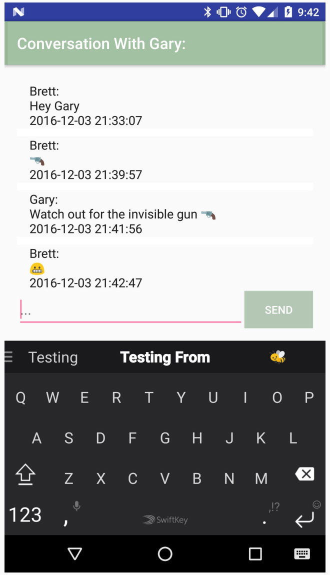
Client Security

- Authentication
  - JWT received upon login
    - Used for authentication when sending messages



Client Security

- Confidentiality
  - Messages Encrypted using AES and CTR mode
    - IV || cipherText || tag || RSAencrypt(k_e || k_i)
    - New k_e and k_i is generated for each message sent
- Integrity
  - Tag created using HmacSHA256
    - When message is received, calculates HMAC(cipherText, k_i), then compares it to the tag received in the message


Conversation With Gary:

Brett:
Hey Gary
2016-12-03 21:33:07

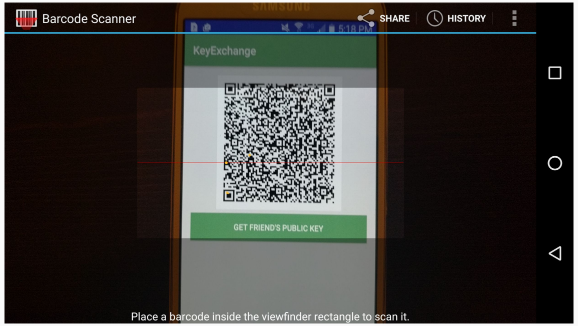
Brett:
2016-12-03 21:39:57

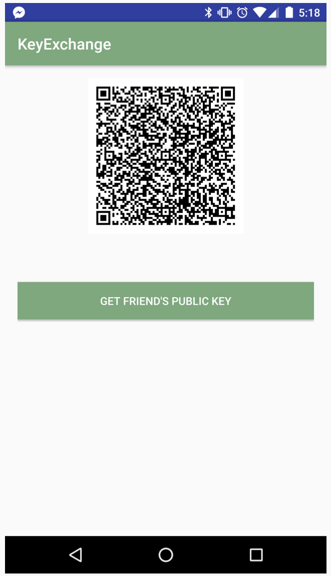
Gary:
Watch out for the invisible gun
2016-12-03 21:41:56

Brett:
2016-12-03 21:42:47

Client Security

- Public Key Exchange
  - ZXing library for QR code scanning
    - Generate QR with Public Key
    - Scan friend's Public Key





Difficulties

- No previous experience configuring web servers
- Client only shows messages sent to me in conversation
    - Can’t decrypt messages I send (because they are encrypted with someone else’s public key)