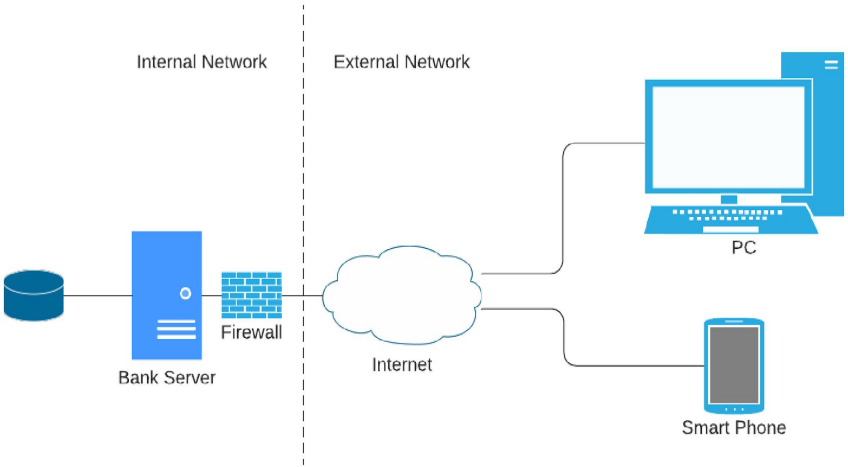
SIRS - Smartphone as a security token

Grupo 31:

- Samuel Vicente, 87704
- Ricardo Caetano, 87699
- Tomás Silva, 83862

General Architecture



Bank Server:

- Django, SQLparse, SQLite, python cryptography
- TLS, Custom Protocol
- Hosted on Digital Ocean with Apache 2
- Let's Encrypt with cert-bot for TLS Certificates

SmartPhone:

- Android Biometric Prompt
- Custom Protocol
- Java Security
- Android KeyStore

Firewall:

- Iptables used for DDoS mitigation

Key Distribution and Management Mechanism

Key Management Mechanism:

- We created a self-signed CA to sign the Server Certificate for application communication
- CA Certificate saved as PEM on assets of application, for certificate validation on the client side
- Client Key Pair is saved using Android KeyStore, requiring user authentication to use and with Sign, Verify, encrypt and decrypt capabilities
- Server TLS Certificate generated using cert-bot, saved on a root user accessible only directory

Key Distribution:

- Server sends certificate to client App
- Client comes preloaded with root CA to use to validate Server Certificate.
- Client sends public key to Server

General Protocol/Architecture: Apache Web Server

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
        ServerAdmin tomas.r.silva@tecnico.ulisboa.pt
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        Alias /static /home/sirs/SIRS/webApp/sirsbank/static
        <Directory /home/sirs/SIRS/webApp/sirsbank/static>
                Require all granted
        </Directory>

        <Directory /home/sirs/SIRS/webApp/sirsbank/>
                <Files wsgi.py>
                        Require all granted
                </Files>
        </Directory>

        WSGIScriptAlias / /home/sirs/SIRS/webApp/sirsbank/sirsbank/wsgi.py
        WSGIDaemonProcess bank_app python-path=/home/sirs/SIRS/webApp/sirsbank python-home=/home/sirs/SIRS/webApp/env
        WSGIProcessGroup bank_app

        ServerName sirsbank.tk
        SSLCertificateFile /etc/letsencrypt/live/sirsbank.tk/fullchain.pem
        SSLCertificateKeyFile /etc/letsencrypt/live/sirsbank.tk/privkey.pem
        Include /etc/letsencrypt/options-ssl-apache.conf
</VirtualHost>
</IfModule>
```

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

In the Apache Web Server: To secure the messages exchanged between the server and the browser we use HTTPS. We used Certbot to generate certificates to Let’s Encrypt.

Custom Protocol: Register Operation

Before communication:

1. User launches client app, for the first time, selects register option
2. Client app sees that no key pair is available so creates one (RSA 2048), protected by user authentication with biometrics and stored on KeyStore
3. Client inputs 6 digit token shown on the register page of the web app
4. Protocol is executed using the created key pair and if successful client public key is associated with account on the server side, registering that phone.

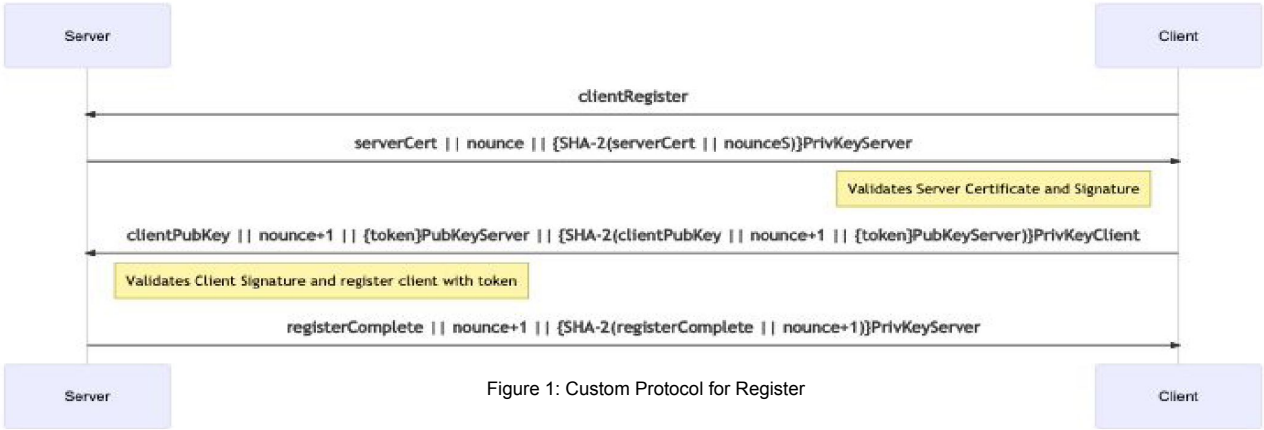


Figure 1: Custom Protocol for Register

Custom Protocol: Login / Authorization Operation

Before communication:

1. User launches client app, selects authentication option
2. Client app sees that there is a key pair is available and asks user for biometric authentication to use that key pair
3. With keys available protocol is executed

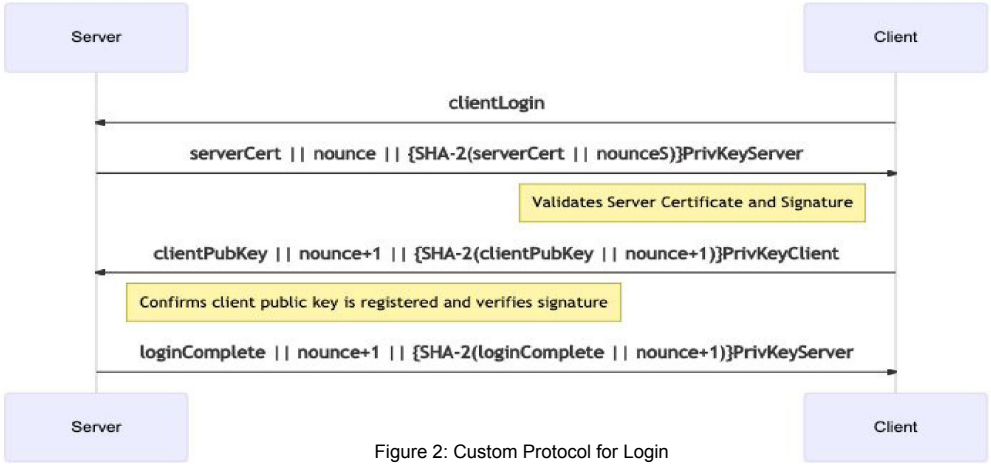


Figure 2: Custom Protocol for Login

Security details

- Passwords must be secure (at least 8 characters long and include at least one uppercase, one lowercase letter, a number and a special character)
- Passwords are hashed with an unique salt
- SQL Injection, XSS and CSRF protection
- Sessions expire in 30 minutes
- Registration tokens are only valid for 5 minutes
- Firewall blocks all invalid TCP packets and ICMP packets to prevent DDoS and ping-of-death attacks

Demonstration/Use case

Link: sirsbank.tk

1. The user starts by signing up in the web application.
2. The user then must use his smartphone and input the generated code present on the screen.
3. The user now has access to his account, to perform critical operations the user must authorize via the smartphone

Topics that we can improve

- Waiting for the 2FA to reveal if the attempted password is correct, preventing brute force or dictionary attacks
- Dictionary attacks protection by forcing the user to choose an uncommon password and add verification to ensure that the password is not similar to the user’s email/name
- Cross location of authentication (phone) and location of login(webapp)