UNIVERSIDADE FEDEAL DE SANTA CATARINA
INE-DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

PROCESSO SELETIVO LABSEC

PEDRO HENRIQUE TAGLIALENHA
(22203674)

DESAFIO APLICATIVO MÓVEL FLUTTER

FLORIANÓPOLIS
FEVEREIRO DE 2023

1.INTRODUÇÃO

Neste relatório apresenta-se as etapas de construção do projeto de um aplicativo móvel capaz de gerar uma lista de dispositivos BLE e um par de chaves RSA, utilizadas para a assinatura digital da lista e realizar a verificação de autenticidade da assinatura. O par de chaves é composto por uma chave pública e uma chave privada, que são usadas para a cifragem e decifragem de informações. A assinatura digital é um método utilizado para garantir a autenticidade e integridade de documentos eletrônicos, utilizando a criptografia assimétrica para criar uma assinatura digital que pode ser verificada pela chave pública. Este aplicativo se propõe para colocar em prática esses conceitos.

2. DESCRIÇÃO DO APLICATIVO

A tela inicial do aplicativo (Figura 1) é a primeira a ser apresentada ao usuário ao abrir o App e possui quatro botões que são responsáveis pela navegação entre as telas. O primeiro botão é responsável por levar o usuário para a tela de Scanner de Dispositivos BLE (Figura 2), que permite a busca de dispositivos Bluetooth Low Energy próximos. Já o segundo botão leva o usuário para a tela de Geração de Chaves RSA (Figura 3), onde é possível gerar pares de chaves assimétricas para serem utilizadas na assinatura digital.

O terceiro botão é responsável pela navegação para a tela de Assinatura Digital (Figura 5), permitindo a criação de uma assinatura para a lista de dispositivos BLE, criada na primeira tela, utilizando as chaves geradas anteriormente. Por fim, o quarto botão leva o usuário para a tela de Verificação de Assinatura Digital (Figura 6), onde é possível verificar a integridade da assinatura criada na tela de Assinatura Digital, utilizando a chave pública correspondente à chave privada utilizada para criar a assinatura.

Figura 1: Menu

Figura 2: BLE

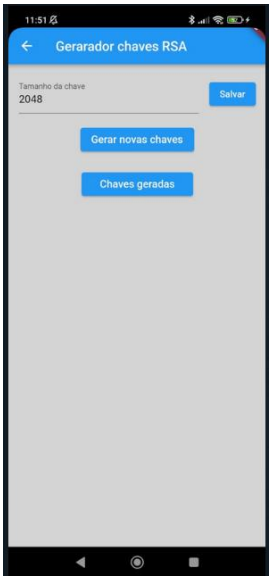

Figura 3: Gerador RSA

Figura 4: Chaves RSA

A tela de Scanner de Dispositivos BLE (Figura 2) é responsável por permitir a busca de dispositivos Bluetooth Low Energy. Ao pressionar o botão

com o ícone de Bluetooth, o aplicativo inicia o processo de varredura em busca de dispositivos BLE disponíveis nas proximidades. As informações dos dispositivos encontrados são exibidas na interface do usuário, juntamente com a data e hora do último scan.

A tela de Geração de Chaves RSA (Figura 3) é responsável por gerar pares de chaves assimétricas utilizadas na assinatura digital. Nessa tela, o usuário informa o tamanho da chave que deseja gerar e em seguida pressiona o botão "Gerar Chaves". Após a geração das chaves, os valores das chaves são armazenados e são disponibilizadas para outras telas do App. O usuário pode visualizar os valores das chaves geradas ao pressionar o botão "Chaves Geradas", que exibe os valores das chaves pública e privada (Figura 4).

A tela de Assinatura Digital (Figura 5) é responsável por receber a lista de dispositivos BLE obtida pelo Scanner e os valores das chaves RSA gerados pela tela de Geração de Chaves. Ao pressionar o botão "Assinar", o aplicativo realiza o processo de assinatura digital dos dados, utilizando a chave privada para gerar a assinatura. O resultado da assinatura digital é exibido na interface do usuário, permitindo que o usuário visualize os dados da assinatura gerada.

A tela de Verificação de Assinatura (Figura 6) é responsável por receber a assinatura digital gerada na tela de Assinatura Digital e a chave pública RSA correspondente. Ao pressionar o botão "Verificar", o aplicativo realiza o processo

de verificação da assinatura digital utilizando a chave pública. O resultado da verificação da assinatura é exibido na interface do usuário, com um texto vermelho "Assinatura não é válida" caso a assinatura não seja válida (Figura 6), ou um texto verde "Assinatura é válida" caso a assinatura seja válida (Figura 7).

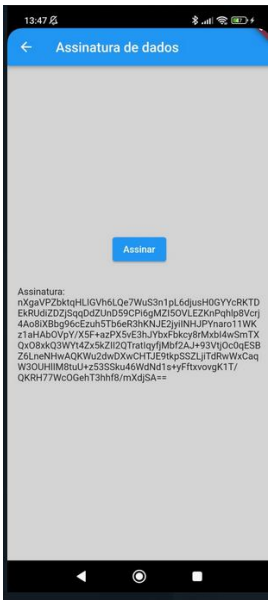

Figura 5: Assinatura Figura 6: Verificador de assinatura Figura 7: Assinatura válida

Para verificar a funcionalidade da verificação gera-se uma lista BLE, um par de chaves e realiza-se a assinatura digital. Ao realizar a verificação, caso a assinatura esteja válida, isso indica que a funcionalidade está operando corretamente. No entanto, é importante destacar que ao gerar um novo par de chaves sem realizar nova assinatura, a verificação retornará um resultado inválido. Da mesma forma, ao se gerar uma nova lista sem realizar nova assinatura, também se obtém um resultado inválido.

3. PACOTES UTILIZADOS

Para o desenvolvimento do App utilizou-se as bibliotecas e pacotes descritos no Quadro 1.

Quadro 1. Bibliotecas e pacotes utilizados

'dart:async'
'dart:convert'
'package:intl/intl.dart'
'package:flutter/material.dart'
'package:fast_rsa/fast_rsa.dart'
'package:provider/provider.dart'
'package:flutter_blue/flutter_blue.dart'

O pacote 'dart:async' contém as classes para trabalhar com programação assíncrona no Dart, como Future, Stream e Completer.

O pacote 'dart:convert' fornece funções para converter entre objetos Dart e codificações de dados comuns, como JSON, UTF-8 e base64.

O pacote 'package:intl/intl.dart' fornece ferramentas para trabalhar com formatação de datas, números e moedas em diferentes idiomas e regiões.

O pacote 'package:flutter/material.dart' fornece as classes e widgets para construir a interface do usuário.

O pacote 'package:fast_rsa/fast_rsa.dart' é uma biblioteca para trabalhar com criptografia RSA no Flutter. Ele permite gerar chaves RSA, criptografar e descriptografar dados usando a chave pública e privada.

O pacote 'package:provider/provider.dart' é uma biblioteca para gerenciamento de estado em Flutter. Ele fornece a classe Provider para injetar objetos em widgets filhos, permitindo que esses widgets acessem e atualizem o estado do aplicativo de forma reativa.

O pacote 'package:flutter_blue/flutter_blue.dart' é uma biblioteca para trabalhar com Bluetooth Low Energy (BLE) em Flutter. Ele fornece classes e métodos para descobrir e conectar dispositivos BLE, bem como para enviar e receber dados através de uma conexão BLE.

4. EXECUÇÃO DO DISPOSITIVO MÓVEL

Para executar o projeto Flutter siga os seguintes passos:

1. Instale o Flutter SDK em sua máquina;
2. Configure o ambiente: configure o PATH do flutter em seu sistema operacional;
3. Instale as dependências do projeto: execute o comando “flutter pub get” no terminal;
4. Execute o projeto: execute o projeto com o comando “flutter run”;

Após esses 4 passos já é possível executar o projeto flutter em um computador ou em um emulador. Porém para executar o App em um celular Android é necessário ativar o modo desenvolvedor indo nas configurações > sobre o celular e clicar 7 vezes no botão de “Build”. Feito isso, vá para as configurações de desenvolvedor e ative depuração por USB. Então conecte o seu dispositivo ao computador utilizando um cabo USB e selecione o celular ao executar o projeto para instalar o aplicativo no celular.

Outra opção é baixar o aplicativo utilizando o .apk por meio do seguinte link:
https://drive.google.com/file/d/1pHl6aK8ejne6QVbCdDaOMCsPf510LLAy/view?usp=sharing

5. DESAFIOS ENCONTRADOS

Durante o processo de desenvolvimento do projeto, enfrentei diversas dificuldades que exigiram várias tentativas e experimentações para serem superadas. A primeira dificuldade foi aprender e aplicar adequadamente o Flutter, já que este foi o meu primeiro projeto desenvolvido utilizando esse framework. Outro desafio encontrado foi a implementação da página de assinatura da lista BLE. Especificamente, para entender como acessar e tratar os dados de outras classes para que a assinatura digital funcionasse corretamente. Tentei várias opções, porém a que melhor se adequou às necessidade do projeto foi o pacote “Provider”. Além disso, tive que aprender a manipular e processar os dados para gerar a assinatura utilizando o algoritmo RSA e PKCS1v15. Foram várias tentativas e implementações do código BLE e de assinatura para finalmente consegui fazer a assinatura funcionar corretamente. Apesar desses problemas, foi possível superar a maioria das dificuldades com a ajuda de pesquisas, tutoriais online e documentações.

Porém, durante o processo de desenvolvimento do aplicativo, encontrei uma dificuldade que não consegui superar. Na página de dispositivos BLE, implementei um botão para realizar um novo scan de dispositivos, mas percebi que ele só funcionava corretamente quando o aplicativo era reiniciado, isto é, ele escaneava apenas uma vez por reinício do App. Tentei várias soluções, incluindo mudar a estrutura das classes, mudar os métodos utilizados, modificar o Provider, implementar novas bibliotecas e mudar o comportamento do botão, mas não consegui resolver o problema completamente. Para realizar o teste da verificação da assinatura contornando esse problema, podemos realizar uma

assinatura com a lista BLE vazia, ao verificarmos a assinatura retornara como válida, porém, ao gerarmos uma lista nova e verificarmos a assinatura anterior, ela retornará invalida. Isso se dá pelo fato que no processo de verificação se faz comparando o resultado da decriptação com a mensagem original assinada.

O desenvolvimento do App foi um processo desafiador, mas me permitiu aprimorar minhas habilidades de resolução de problemas e aprender mais sobre a arquitetura e desenvolvimento de aplicativos móveis, e me proporcionou uma oportunidade para aprender e crescer profissionalmente.

6.CONCLUSÃO

Para o desenvolvimento do aplicativo móvel foi preciso trabalhar com os conceitos de assinaturas digitas utilizando pares de chaves RSA o que possibilitou aprimorar meus conhecimentos relacionadas à criptografia e segurança da computação. Ademais, o desenvolvimento do App oportunizou melhorar minhas habilidades com o Framework Flutter, que é uma plataforma de desenvolvimento de aplicativos móveis versátil e muito utilizado no mercado de trabalho. Dessa forma, o projeto se mostrou uma excelente oportunidade para desenvolver habilidades técnicas e ampliar conhecimentos na área de tecnologia da informação.