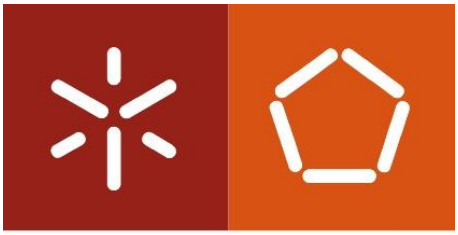

Universidade do Minho

Departamento de Informática

Mestrado em Engenharia Informática

Criptografia e Segurança de Informação

Engenharia de Segurança
Projeto de Desenvolvimento 1 - Cofre Digital

Grupo Nº 3

Ariana Lousada (PG47034) Luís Carneiro (PG46541)
Rui Cardoso (PG42849)

2 de maio de 2022

Resumo

O presente documento descreve sucintamente os objetos de avaliação e de análise ao longo do primeiro projeto de desenvolvimento inserido na unidade curricular Engenharia de Segurança, pertencente ao perfil de Criptografia e Segurança da Informação inserido no Mestrado em Engenharia Informática da Universidade do Minho.Este projeto teve como principal objetivo a implementação de um cofre digital para armazenamento e acesso específico a documentos digitais.

Conteúdo

Capítulo 1

Introdução

1.1 Contextualização

Com o crescente número de ataques informáticos que se observa atualmente, é de extrema importância desenvolver aplicações que sejam seguras para os utilizadores, capazes de proteger os seus dados contra terceiros.

Existem já diversas aplicações e sistemas nos quais o cidadão normal necessita na sua vida: Aplicações de bancos, serviços de saúde, entre outras. A exposição indevida deste tipo de dados, caso não sejam protegidos adequadamente, pode levar a consequências muito sérias e prejudiciais para a vida do cidadão.

Para que este tipo de falha não aconteça, é importante definir estritamente o acesso a estes dados críticos.

1.2 Objetivos do Projeto

O principal objetivo deste projeto consiste na implementação de um cofre digital, no qual seja possível o armazenamento de documentos digitais, assim como a atribuição de acesso a apenas aos respetivos titulares.

O depósito do documento deve ser feito por um indivíduo(depositante) que deverá definir os titulares do documento que deposita.

O fornecimento do documento deve ser feito apenas quando parte das entidades de partilha de segredo se encontrem "presentes"(implementado através do esquema de partilha de segredo de Shamir). Por consequência, um documento não deve ser acedido por terceiros que não tenham autorização para tal.

1.3 Estrutura do Relatório

O presente documento encontra-se dividido nas seguintes secções:

- Conceção/Desenho da solução (2): Construção da solução e respetiva implementação. Exposição das várias decisões tomadas pela equipa de trabalho.
- Codificação e Testes (3): Elaboração de testes de depósito/fornecimento de documentos do cofre

digital; Exposição de alternativas para possível melhoria da solução e de problemas de implementação.

Capítulo 2

Conceção/Desenho da solução

2.1 Entidades

As entidades principais são o cofre (*vault*), a pessoa (*person*) e o documento (*document*).

O cofre tem uma lista de registos (*vault records*), que são compostos pelo nome do ficheiro (para uma identificação fácil), o hash do documento calculado pelo cofre, o documento cifrado pela cifra simétrica AES, assim como o vetor de inicialização usado por esta, o depositante e se o esquema de Shamir foi usado.

A pessoa tem um nome, as próprias chaves pública e privada (assim como os nomes dos ficheiros onde estão guardadas) e uma lista de registos necessária para aceder a documentos depositados. Nesta lista estão presentes o nome do ficheiro, o hash do documento, a chave cifrada devolvida pelo cofre (resultado de aplicar uma cifra RSA com a chave pública do depositante à chave usada para cifrar o documento) e o vetor de inicialização usado na cifra AES.

O documento tem um nome e conteúdo.

Das outras entidades secundárias destacam-se o AES, que fornece funções para cifrar/decifrar conteúdo com uma chave, RSA, que cifra com a chave pública e decifra com a privada, SHA-256 para calcular o hash do documento e *String to Int*, para converter o output da cifra RSA para inteiro (para esta cifra poder ser distribuiída no esquema de Shamir).

2.2 Depósito do documento

Apenas é possível depositar um documento para ser visto pelo depositante. Por outras palavras, não foi possível disponibilizar o esquema de Shamir, por razões que irão ser explicadas mais à frente.

Primeiro, é preciso criar uma instância do cofre. Esta vai, através do openssl, criar uma chave privada e pública, assim como criar um certificado *root*, disponível para assinar as *certificate requests* dos seus utlilizadores aquando registo.

Depois, a pessoa que deposita o documento é criada (mais uma vez, o openssl trata de criar as suas chaves pública e privada) e registada no cofre, o que permite associar a sua identidade pessoal à sua chave pública.

De seguida, o depósito propriamente dito é feito, sendo que o cofre e o depositante guardam os dados descritos no enunciado, isto é, o par hash, documento cifrado para o cofre e o par hash, chave de decifra do documento (cifrado com a chave pública do depositante) para o mesmo.

2.3 Fornecer documento

O depositante tem de apenas indicar o nome do documento para o aceder. No *backend* do programa, é procurado o seu nome nos registos do mesmo. Caso ele exista, o pedido é feito ao cofre. Aqui, para além de ser validado se o ficheiro existe, tanto a integridade do documento como a identidade do depositante vai ser verificada (através de um pedido openssl que verifica o seu certificado), de modo a garantir que apenas este pode ver o documento. Se tudo estiver de acordo, o documento é devolvido. Caso contrário, uma mensagem de erro é emitida.

Capítulo 3

Codificação e Testes

3.1 Testes realizados e Resultados

```
User Luis has certificate name VdSFmT6FNWnIxw==
./scripts/openssl_create_user_keys.sh VdSFmT6FNWnIxw== pubKey_VdSFmT6FNWnIxw== privKey_VdSFmT6FNWnIxw==
User Rui has certificate name sU7QkKZ8H1taUQ==
./scripts/openssl_create_user_keys.sh sU7QkKZ8H1taUQ== pubKey_sU7QkKZ8H1taUQ== privKey_sU7QkKZ8H1taUQ==
./scripts/openssl_verify.sh VdSFmT6FNWnIxw==
./certificados/VdSFmT6FNWnIxw==.crt: OK
O documento original é testeeeee
Erro: O Rui nao depositou um documento com o nome test
./scripts/openssl_verify.sh VdSFmT6FNWnIxw==
Não é possível autenticar utilizador...
```

Figura 3.1: Teste de depósito do documento

Nas primeiras quatro linhas do teste da figura anterior são gerados os certificados de dois utilizadores distintos. Neste caso em particular, o utilizador `Luis` depositou um documento `teste.txt` com conteúdo `testeeeee`.

Na quinta linha, é verificada a validade dos certificados gerados para cada utilizador. Uma vez que se obteve a resposta `OK`, pode-se concluir que ambos os certificados são válidos.

Contudo, apenas o primeiro utilizador(neste caso o `Luis`) tem acesso ao conteúdo do ficheiro. Uma vez que o utilizador `Rui` não depositou qualquer documento com o nome *test*, o acesso não lhe é atribuído.

3.2 Problemas de Implementação

No desenvolvimento do presente projeto, a equipa de trabalho deparou-se com um problemas de utilização excessiva de memória que influenciou negativamente a *performance* do programa. Após várias análises ao código feitas pela equipa de trabalho, concluiu-se que este uso excessivo de memória provém da utilização de `BigInteger` em algumas variáveis.

Contudo, o grupo não conseguiu encontrar uma solução para o problema. Uma vez que o número de dígitos do segredo é de tamanho significativo, era necessário utilizar um formato do tipo do `BigInteger`.

Apesar da partilha de segredo de Shamir funcionar para segredos de tamanho reduzido, o tempo de execução do programa vai aumentando bastante a partir do momento que se utilizem segredos de 7 ou mais dígitos.

Capítulo 4

Conclusão

Com o desenvolvimento deste projeto a equipa de trabalho conseguiu explorar técnicas de partilha de segredo como o Shamir, o que possibilitou a compreensão da importância deste tipo de métodos para garantir a integridade, privacidade, confidencialidade dos documentos neste de tipo de aplicação.

Adicionalmente, proporcionou uma experiência prática na criação e geração de chaves e certificados, assim como da sua verificação, de modo a associar com segurança a identidade de um utilizador com as suas chaves pública e privada.

Capítulo 5

Referências

- https://www.oodlestechnologies.com/blogs/how-to-generate,-use-and-store-public-and-private-keys-in-java/
- https://stackoverflow.com/questions/17322002/what-causes-the-error-java-security-invalidkeyexception-parameters-missing
- https://www.devglan.com/java8/rsa-encryption-decryption-java
- https://www.baeldung.com/java-rsa
- https://stackoverflow.com/questions/11410770/load-rsa-public-key-from-file