Mestrado em Engenharia Telecomunicações e Informática

Cryptography – Designing a Security Protocol

Segurança em Redes e Sistemas de Informação

Docente: Carlos Serrão

Alexandre Rodrigues, nº 105260

Loice Sitole, nº 123848

Maria Vinheiras, nº 105563

Mariana Gonçalves, nº 127524

Ricardo Gouveia, nº 105062

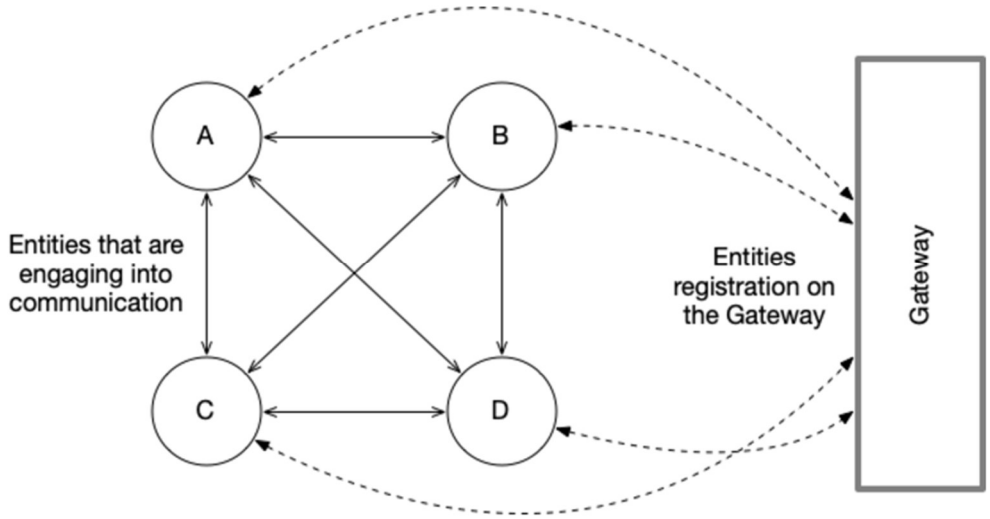
Vasco Tavares, nº 104808

Índice

Atividade 1 - Especificação e desenho de um protocolo seguro

Componentes do sistema

Na seguinte figura, estão representados os componentes do sistema que será implementado ao longo deste trabalho.



-Entidade Alice (A)

-Entidade Bob (B)

-Entidade Charlie (C)

-Entidade Daniel (D)

-Gateway (G)

Requisitos do Sistema

Nesta secção, são apresentados os requisitos fundamentais para a criação de um protocolo seguro que permita a comunicação cifrada, autenticada e descentralizada entre várias entidades. Este sistema foi concebido com base nos princípios e mecanismos de criptografia estudados nesta unidade curricular, utilizando tecnologias que asseguram a sua estabilidade.

A comunicação entre as entidades deverá respeitar os seguintes requisitos:

- O Gateway deve ser utilizado como um ponto central de confiança entre as diferentes entidades de comunicação;

- O sistema de comunicação é totalmente descentralizado (P2P) e privado. As mensagens trocadas são enviadas diretamente aos destinatários sem passar por um servidor centralizado;
- Antes de iniciar a comunicação, todas as entidades precisam autenticar-se mutuamente (deve ser possível que mais de duas entidades participem numa conversa);
- Deve existir um mecanismo que permita às entidades renegociar os parâmetros de segurança entre si e renovar as chaves;
- Todas as mensagens trocadas devem garantir a sua confidencialidade e integridade.

A criação deste sistema requer a utilização de métodos avançados de criptografia, como RSA para a criação de pares de chaves (pública e privada), e AES ou outros algoritmos para a cifragem das mensagens. O uso de certificados digitais emitidos pelo Gateway garantirá a confiança entre as entidades e permitirá a validação mútua das chaves públicas.

Por fim, é fundamental que o design e implementação tenham como base os princípios de segurança já conhecidos, incluindo a minimização de vulnerabilidades, a prevenção de ataques e a garantia de um desempenho favorável a uma comunicação simultânea eficiente entre entidades.

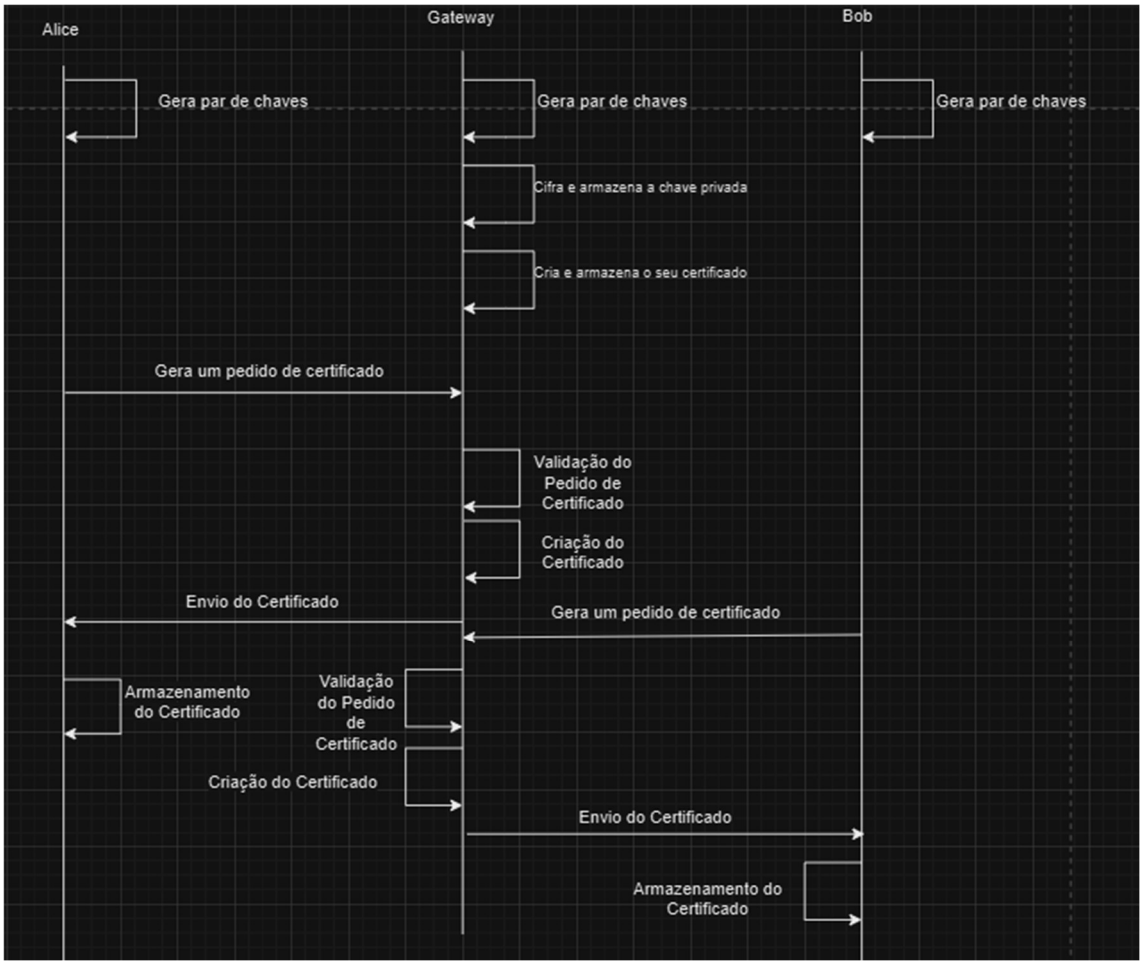
Estes requisitos serão posteriormente demonstrados na especificação do protocolo e na sua implementação usando OpenSSL.

Inicialização do Gateway e Entidades

Na seguinte figura é possível observar o processo de inicialização do Gateway e das entidades Alice e Bob:

1. Todos os elementos geram o seu próprio par de chaves (Pública e Privada) usando o RSA.
2. O Gateway cifra e armazena a sua chave privada.
3. O Gateway cria e armazena o seu certificado.
4. A entidade Alice envia um pedido de certificado ao Gateway.
5. O Gateway valida o pedido de certificado.
6. O Gateway cria o certificado da Alice.
7. O Gateway envia o certificado da Alice para a Alice e ela armazena-o.
8. A entidade Bob envia um pedido de certificado ao Gateway.
9. O Gateway valida o pedido de certificado.
10. O Gateway cria o certificado da Bob.
11. O Gateway envia o certificado da Bob para a Bob e ele armazena-o.

Existe ainda uma entidade Charlie e uma entidade Daniel que realizaram os mesmos pedidos à Gateway que a Alice e o Bob. Decidimos não representar no esquema para não ficar mais repetitivo e confuso.

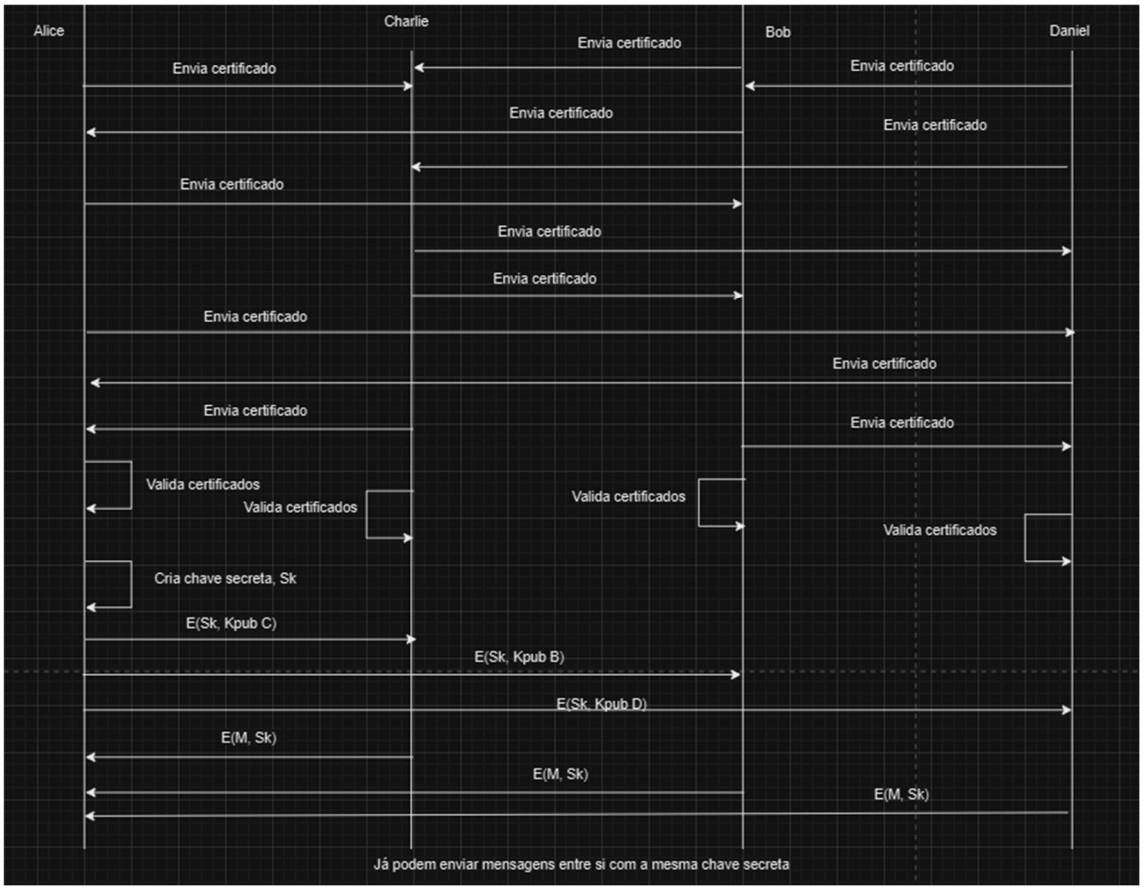


Comunicação entre Entidades

Na seguinte figura observa-se a troca de mensagens entre as duas entidades:

1. As entidades enviam uma a outra os seus certificados.
2. As entidades validam os certificados recebidos.
3. A Alice cria uma chave secreta, Sk.
4. A Alice encripta a chave secreta Sk com a chave pública do Bob, do Charlie e do Daniel e envia as mensagens para as respetivas entidades.
5. Bob, Charlie e Daniel desencriptam a mensagem recebida usando a sua chave privada e me envia uma mensagem com a chave secreta estabelecida.

Agora, as entidades podem enviar mensagens com a chave secreta.



Atividade 2 - Implementação do protocolo seguro usando OpenSSL

Comunicação Segura

Objetivo:

Assegurar que todas as mensagens enviadas entre as partes estejam protegidas contra intercetação por terceiros, garantindo a privacidade das informações transmitidas.

Descrição e Implementação:

- Utilização do algoritmo AES (Advanced Encryption Standard) para cifrar mensagens com uma chave simétrica, garantindo a confidencialidade.
- A troca inicial de chaves será feita através de um algoritmo assimétrico como RSA, que proporciona segurança adicional na distribuição de chaves.
- A assinatura digital será usada para verificar a autenticidade de cada mensagem cifrada.

Impacto na Segurança:

A utilização de cifragem forte e assinaturas digitais garante que as mensagens sejam decifradas e verificadas apenas por quem é permitido, evitando ataques de intercetação e alteração.

Comandos para o protocolo

Num ambiente que suporte a ferramenta OpenSSL, começamos pela configuração do nosso protocolo, o Gateway necessita de gerar o seu par de chaves. Antes de se tornar uma entidade geradora de certificados para as restantes entidades, o próprio necessita de gerar o seu próprio certificado.

```
mkdir private certs newcerts crl
```

Criamos as quatro diretórias para colocar as respetivas informações.

```
touch index.txt
```

```
echo '01' > serial
```

Criamos um ficheiro de texto "index" vazio e escrevemos "01" num ficheiro "serial"

```
openssl req -config ./openssl.cnf -new -x509 -extensions v3_ca -
keyout private/chavepriv.key -out certs/gateway.crt -days 1095
```

Este comando permite-nos criar um certificado com uma estrutura X.509 e utilizar a extensão v3_ca do ficheiro *openssl.cnf* e assim como escolher o local onde armazenar a chave privada e o certificado. Este certificado expirará em 3 anos.

Ao prosseguir com este comando aparecerá um prompt a solicitar informação como, o nome do país, cidade, o nome da organização e *common name* e opcionalmente o email da entidade que está a requisitar o certificado.

```
Country Name: PT
```

```
State or Province Name: Lisboa
```

```
Locality Name: Lisboa
```

```
Organizational Unit Name: Protocolo_Seguro
```

```
Organization Name: Protocolo_Seguro
```

```
Common Name: Gateway
```

Com esta inicialização do Gateway realizada, este estará pronto para assumir o papel de entidade Certificate Authority.

Agora, para fazer setup dos agentes é necessário:

Com fizemos no Gateway, criamos à mesma um diretório para alocar cada tipo de informação. E daí prosseguimos para a produção do certificado.

```
openssl req -config ./openssl.cnf -new -keyout
./private/agente.key -out agente.csr -days 365
```

Este comando gera um pedido para um certificado digital de uma das entidades para a gateway. O pedido vai conter a chave pública da entidade e a sua assinatura digital. A gateway valida o pedido através da assinatura digital e extraí de lá a chave pública da entidade. Para a gateway conseguir visualizar a informação que necessita corre o seguinte comando:

```
openssl req -in ./agente.csr -noout -text
```

Que permite criar e guardar o certificado requerido pela entidade. O certificado irá conter a chave pública da entidade assinada pela gateway.

```
openssl ca -config ./openssl.cnf -policy policy_anything
-out certs/agente.crt -infiles agente.csr
```

Com este comando a gateway gera o certificado propriamente dito e no final da operação o "agent.csr" irá conter o certificado.

```
openssl x509 –noout –text -in agente.crt
```

Assim a entidade irá ver o conteúdo do certificado enviado pela gateway bem como o certificado da gateway que também lhe foi enviado.

Após todas as entidades passarem pelo processo de adquirir um certificado, podem começar a comunicar com outras entidades, enviando primeiramente o certificado adquirido para comprovar a sua identidade, tendo esse de ser validado pela entidade que o recebe através do comando:

```
openssl verify -CAfile gateway.crt agente2.crt
```

Após a verificação do certificado vai verificar de acordo com o certificado da Gateway:

```
openssl verify -CAfile gateway.crt agente1.crt
```

Para poderem começar a comunicar entre si uma das entidades irá gerar uma chave secreta que irá encriptar numa mensagem com a chave publica de cada entidade de destino e enviar as mensagens para as respetivas entidades.

```
openssl rand -hex 16 > sk.key
```

Mas antes terá de extrair a chave publica das entidades dos seus certificados através de:

```
openssl x509 –pubkey –noout -in agente2.crt > agente2key.pub
```

De seguida já s epode encriptar a chave secreta com a chave pública da entidade a que se destina a mensagem:

```
openssl pkeyutl –encrypt -inkey agente2_key.pub -pubin -in
sk.key -out sk.key.enc
```

“sk.key.enc” irá conter a chave secreta encriptada com a chave pública da entidade de destino

Do lado da entidade de destino, esta irá desencriptar a mensagem com a sua chave privada através de:

```
openssl pkeyutl –decrypt -inkey agente2.key -in sk.key.enc -out
sk.key
```

Assim ambos os agentes partilham uma chave secreta e a partir deste momento podem começar a enviar mensagens (M) encriptadas com essa chave secreta, mas primeiro terá de se encriptar a mensagem M com a chave secreta:

```
openssl aes-256-cbc –e -in M.txt -K
7c171a46d72a741021725082da859625 -out M.enc -base64
```

Quando a entidade recebe a mensagem irá decifrá-la com:

```
openssl aes-256-cbc –d -in M.enc -K
7c171a46d72a741021725082da859625 -out M.txt -base64
```

E assim consegue aceder ao conteúdo do ficheiro “M.txt” que contém a mensagem que se queria encriptar.

A partir de agora os agentes pode comunicar seguramente com a utilização da chave secreta criada anteriormente.

Atividade 3 - Implementação do protocolo seguro usando uma biblioteca criptográfica

Esta implementação foi realizada nos ficheiros anexados, em que o ficheiro README contém informação de como usar o código.

Conclusão

Este trabalho apresentou a especificação e implementação de um protocolo seguro que garante a comunicação entre a Alice, o Bob, o Charlie e o Daniel respeitando a confidencialidade e autenticidade entre as partes. Foram implementados princípios fundamentais de criptografia como o RSA para a troca das chaves e AES para a cifragem das mensagens, tornando possível o desenvolvimento de um sistema robusto que atende os requisitos de segurança e descentralização. A confiança mútua entre as entidades foi assegurada pela emissão de certificados digitais pelo Gateway, que atua como uma Autoridade de Certificação confiável. Além disso, a flexibilidade para renegociação de parâmetros de segurança fortalece a resiliência do sistema frente a ataques e mudanças de contexto.