Engenharia de Telecomunicações e Informática

Segurança em Redes e Sistemas de Informação

Projeto e Implementação de um Protocolo de Segurança Criptográfica

João Rabuge | 98509

Bernardo Assunção | 98616

Hugo Costa | 93910

Docente:

Nuno Filipe Martins de Silveira Reis

Ano curricular: 4º

Semestre: 1º

Semestre **2023/2024**

Índice

1. Introdução

O crescente desenvolvimento e expansão da tecnologia trouxe um número crescente de desafios na área da **segurança da informação**. Uma das áreas onde esses desafios são mais proeminentes é na **comunicação segura** entre **entidades** em redes. Com a necessidade de garantir a **autenticidade**, a **confidencialidade** e a **integridade** das informações trocadas, a **criptografia** tornou-se uma ferramenta indispensável em qualquer sistema de **segurança de informação**.

Neste contexto, este documento apresenta o **design** e a **implementação** de um **protocolo criptográfico seguro** destinado a ser usado em um sistema de chat seguro. O protocolo foi projetado para permitir a **comunicação segura** e **autenticada** entre várias entidades (A, B, C, D, etc) através de um *gateway* central. Este *gateway* atua como um ponto de **confiança central**, facilitando o processo de **autenticação entre as entidades**.

A arquitetura do sistema é **descentralizada** e **privada**, com a **troca de mensagens** ocorrendo diretamente entre as entidades sem passar por um servidor centralizado. Esta abordagem descentralizada tem a vantagem de **reduzir a dependência** de um **único ponto de falha,** aumentando a **resiliência do sistema**.

O protocolo proposto foi projetado para atender a uma série de **requisitos de segurança**. Antes de iniciar a comunicação, todas as entidades precisam de se **autenticar**. Além disso, deve existir um mecanismo que permita às entidades **renegociar os parâmetros de segurança** entre si e renovar as chaves. Por fim, todas as mensagens trocadas devem garantir a sua **confidencialidade** e **integridade**.

Este documento está estruturado da seguinte maneira:

- **Secção 1.** - Introdução do projeto,
- **Seção 2.** - Apresenta a especificação e o design do protocolo seguro proposto,
- **Seção 3.** - Descreve a implementação do protocolo utilizando a ferramenta OpenSSL,
- **Seção 4.** – Plano de resposta a incidentes,
- **Seção 5.** – Conclusão do projeto.

2. Especificações e Design de um Protocolo Seguro

A seção a seguir destaca os detalhes do **design** e da **especificação** do **protocolo seguro** proposto. O protocolo foi concebido tendo em consideração os pri**ncípios fundamentais da criptografia** e da **segurança da rede**, garantindo a **autenticação**, a **confidencialidade**, a **integridade** e a **disponibilidade** das comunicações. O design do protocolo envolve várias etapas, incluindo o **registo da entidade**, a **autenticação**, a **troca de chaves**, a **troca segura de mensagens** e a **renegociação de chaves**.

2.1. Registo de uma Entidade

O primeiro passo na implementação do protocolo seguro é o **registo da entidade**. Neste processo, cada entidade participante se **regista no *Gateway***, que atua como o ponto de **confiança central** no sistema de comunicação. Para garantir a **segurança** na comunicação, cada entidade deve possuir um **par de chaves único**, composto por uma **chave pública** e uma **chave privada**.

O ***Gateway*** é responsável pela **geração do par de chaves para cada entidade**. Utiliza-se um algoritmo de **chave assimétrica**, como o **RSA**, para a geração das chaves. Uma vez geradas, o *Gateway* fornece à entidade o par de chaves: a **chave privada**, que deve ser mantida em **segredo pela entidade**, e a **chave pública**, que pode ser **distribuída livremente**. A **chave privada** será utilizada para **assinar digitalmente** as mensagens e a **chave pública** será utilizada para **verificar as assinaturas**.

O processo de **registo** garante que cada entidade tenha um **par de chaves único**, permitindo assim a **autenticação segura** e a **troca de mensagens** entre as entidades. Além disso, o uso de um sistema de **chave assimétrica** permite o estabelecimento de uma **comunicação segura**, mesmo em um ambiente onde a segurança total não pode ser garantida.

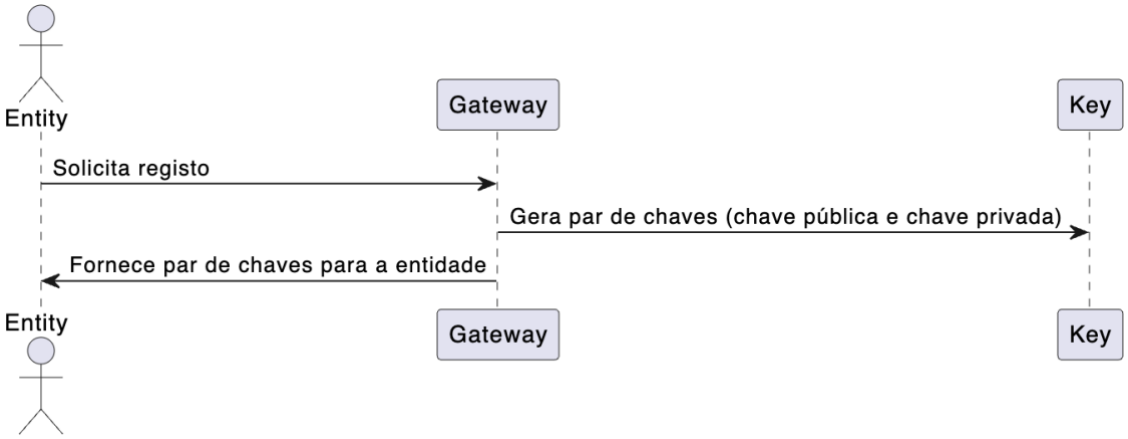

Figura 1. Diagrama de Registo de Entidade.

2.2. Autenticação

Após o **registo bem-sucedido** de uma **entidade** e a **criação do par de chaves**, o próximo passo crucial no protocolo seguro é a **autenticação**. Este processo é essencial para garantir que as entidades envolvidas na comunicação sejam **quem afirmam ser**, prevenindo assim qualquer forma de **usurpação de identidade** ou **tentativas de spoofing**.

A **autenticação** no protocolo é realizada da seguinte maneira:

- **Início da Comunicação**: A entidade A, que deseja iniciar a comunicação com a entidade B, envia uma solicitação ao *Gateway*. Esta solicitação inclui a identidade da entidade A, que é essencialmente uma representação única da entidade, como um nome de usuário ou um ID de entidade.

- **Assinatura da Identidade**: Ao receber a solicitação, o *Gateway* assina a identidade da entidade A usando sua própria chave privada. Esta assinatura atua como um endosso do *Gateway* de que a identidade é válida e autêntica.

- **Verificação da Assinatura**: A entidade B, ao receber a identidade assinada, verifica a assinatura usando a chave pública do *Gateway*. Se a assinatura for

verificada com sucesso, isso confirma que a identidade foi realmente assinada pelo *Gateway* e que a entidade A é autêntica.

- **Resposta à Entidade A**: Após a verificação bem-sucedida, a entidade B responde à entidade A com sua própria identidade assinada. Este processo é semelhante ao descrito acima, com o *Gateway* assinando a identidade da entidade B e a entidade A verificando a assinatura.

- **Autenticação Mútua**: Uma vez que ambas as entidades verificaram com sucesso as identidades uma da outra, a autenticação mútua é alcançada. Isso significa que ambas as entidades podem confiar na identidade uma da outra e iniciar uma comunicação segura.

É importante notar que todo o processo de autenticação depende da segurança do ***Gateway*** e de sua **chave privada**. Se a **chave privada do *Gateway*** for comprometida, a **segurança** de todo o protocolo poderá ser **violada**. Portanto, medidas rigorosas devem ser implementadas para proteger a **chave privada do *Gateway***.

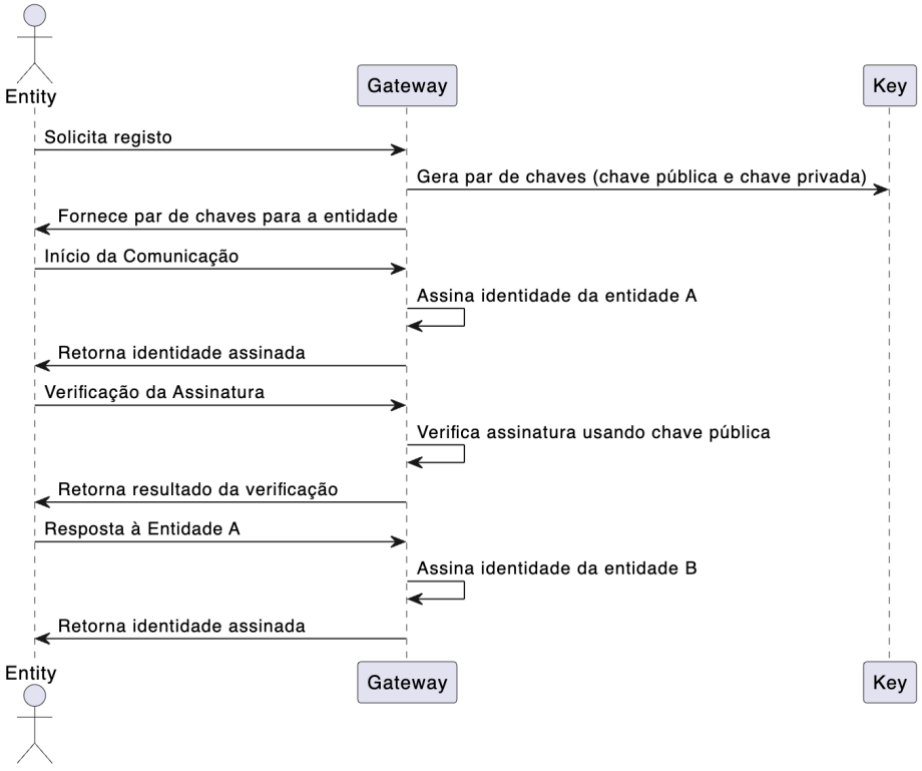

Figura 2. Diagrama de Autenticação de Entidades.

2.2.1.Proteger chave privada do Gateway

Para garantir a segurança da **chave privada** do *Gateway*, devem ser implementadas medidas rigorosas. O **armazenamento da chave** deve ocorrer em um co**fre de chave segura** que é protegido tanto **fisicamente** quanto **digitalmente**. A auten**ticação de dois fatores** pode ser usada para garantir que apenas usuários autorizados tenham acesso à chave. Além disso, rotinas **regulares de backup** e **recuperação de desastres** devem ser estabelecidas para garantir a **disponibilidade** da chave, mesmo em caso de fa**lha do sistema** ou **violação de segurança**.

2.3. Trocas de Chaves

A **troca de chaves** é um processo crítico que ocorre após a auten**ticação mútua bem-sucedida entre as entidades A e B**. Este processo permite que ambas as entidades **concordem com uma chave secreta partilhada** (também conhecida como **chave simétrica**) que será usada para a criptografia e descriptografia de mensagens subsequentes.

O protocolo ***Diffie-Hellman*** é normalmente usado para essa negociação de **chave simétrica**. É um método de **troca de chaves seguro** que permite que duas partes que não têm contacto prévio estabeleçam conjuntamente uma **chave secreta partilhada** por meio de um **canal de comunicação público**.

No contexto deste protocolo, a **entidade A** e a **entidade B** gerem a sua própria **chave privada** e parâmetro público usando o algoritmo *Diffie-Hellman*. Em seguida, eles trocariam a sua **chave pública**. Após receber a **chave pública** um do outro, cada entidade seria capaz de calcular a **chave secreta partilhada** usando a sua própria chave privada e a chave pública recebida da outra entidade.

Esta **chave partilhada** é então usada para criptografar e descriptografar as mensagens trocadas entre a entidade A e a entidade B, garantindo a **privacidade** e a **segurança da comunicação**.

Esta **chave simétrica** é temporária e é descartada após um **período definido**, após o qual uma **nova chave simétrica é negociada** entre as entidades usando o mesmo processo. Isso é conhecido como **renegociação de chaves** e é uma prática padrão em protocolos de segurança para prevenir **ataques de força bruta** de longo prazo.

Para garantir que o protocolo possa ser implementado em **larga escala**, a escalabilidade é uma **consideração crucial**. O protocolo deve ser projetado para manter o **desempenho** sob **alta carga** e deve **minimizar o uso de recursos** para suportar um grande número de entidades. Além disso, o impacto do crescimento no protocolo e na segurança geral do sistema deve ser **avaliado regularmente** para garantir que a **escalabilidade não comprometa a segurança**.

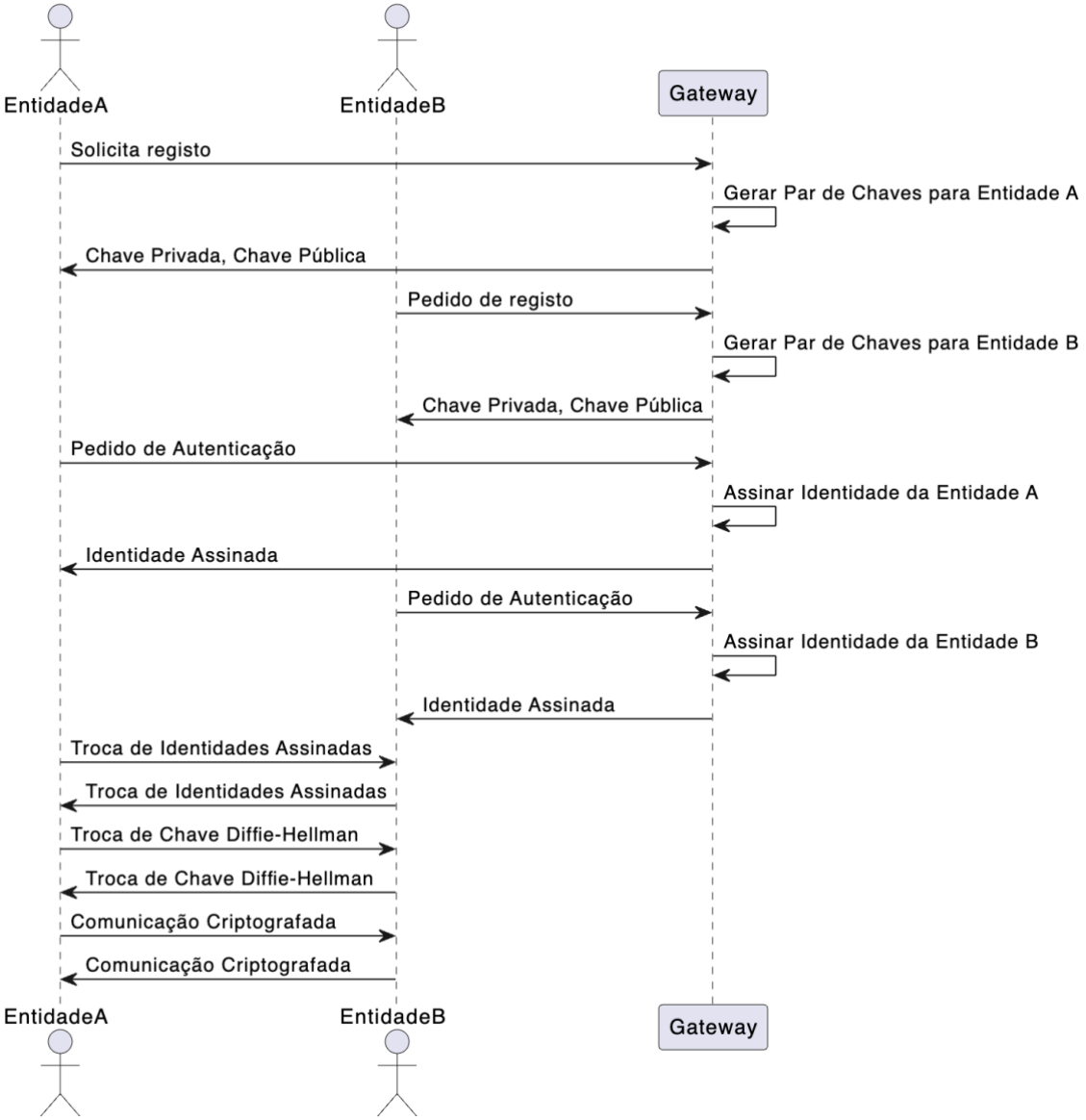

Figura 3. Diagrama de Troca de Chaves entre Entidades.

2.4. Troca Segura de Mensagens

Uma vez **estabelecida a comunicação segura** e **autenticada** e **acordada a chave secreta partilhada** entre as entidades, a **troca segura** de mensagens pode começar. Este processo consiste em duas partes principais: a **criptografia** das mensagens e a **garantia da integridade** das mensagens.

Na **criptografia das mensagens**, as mensagens trocadas entre as entidades são **criptografadas** usando a **chave secreta partilhada**, assegurando a **confidencialidade** das mensagens. A **criptografia simétrica** é escolhida para este propósito devido à sua **eficiência** e **velocidade** em relação à criptografia assimétrica.
O **Advanced Encryption Standard (AES)** é o algoritmo de **criptografia simétrica** escolhido para este propósito. O **AES** é amplamente reconhecido pela sua **segurança** e **eficiência**, tornando-o uma escolha popular para a criptografia de dados sensíveis.

Para garantir a **integridade das mensagens**, ou seja, para garantir que as mensagens não foram alteradas, um **Message Authentication Code (MAC)** é anexado a cada mensagem. O **MAC** é um pequeno bloco de dados que é gerado a **partir da mensagem** e da **chave secreta partilhada.**

Para este propósito, é usado o **código de autenticação** de mensagem baseado em **hash (HMAC)**. O HMAC é um tipo específico de MAC que usa uma função de hash **criptográfico** em **combinação** com uma **chave secreta**.

Portanto, quando uma entidade recebe uma mensagem, ela pode verificar a **integridade** da mensagem **recalculando o HMAC** e **comparando-o com o HMAC** que foi enviado com a mensagem. Se os dois HMACs corresponderem, a entidade pode ter certeza de que a mensagem **não foi alterada**. Se eles não corresponderem, a entidade sabe que a mensagem **foi alterada** e pode tomar as medidas apropriadas.

Em resumo, a troca segura de mensagens é garantida através da criptografia das mensagens para **confidencialidade** e da verificação da integridade das mensagens através do uso de **HMACs**.

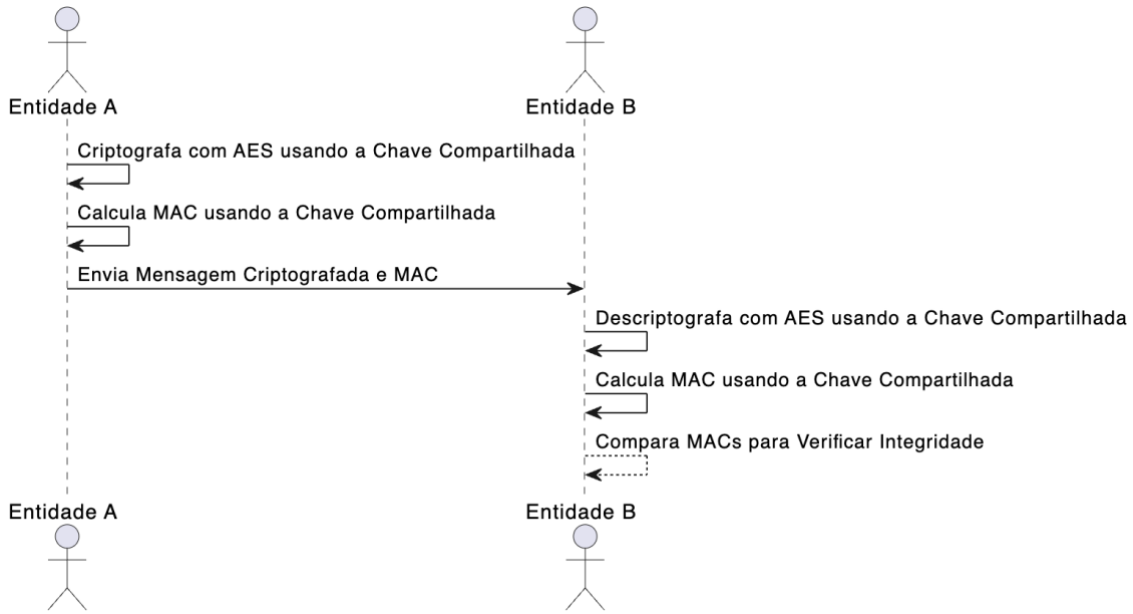

Figura 4. Diagrama de Troca Segura de Mensagens entre Entidades.

2.5. Renegociação de Chaves

Para **aumentar a segurança da comunicação**, é crucial que as **chaves** utilizadas na **criptografia** e **autenticação** das mensagens sejam **renovadas** ou **renegociadas** periodicamente. Isto é especialmente importante em sistemas de comunicação de longa duração, onde o uso prolongado da mesma chave pode aumentar o risco de comprometimento da mesma.

A **renegociação de chaves** envolve a substituição da **chave secreta partilhada** atualmente em uso por uma nova chave. Este processo deve ser realizado de maneira **segura** para garantir que a nova chave não seja **comprometida** durante o processo de **renegociação**.

O protocolo ***Diffie-Hellman***, que foi utilizado para a troca inicial de chaves, pode ser usado novamente para a renegociação de chaves. Este protocolo permite que duas entidades que já têm uma **chave secreta partilhada** negociem uma nova chave secreta partilhada de maneira segura.

O **protocolo de Renegociação de Chaves,** consiste em uma **entidade A** iniciar a **renegociação de chaves**, gerando um novo par de chaves *Diffie-Hellman* e enviando a **chave pública** para a **entidade B**.

Posteriormente, a entidade B recebe a **chave pública** da entidade A, gera o seu próprio par de chaves *Diffie-Hellman*, e envia sua **chave pública** para a **entidade A**.

Após este passo, as entidades A e B usam suas **próprias chaves privadas** e a **chave pública** da outra entidade para calcular a nova **chave secreta partilhada**.

Finalmente, As entidades A e B agora usam a **nova chave secreta partilhada** para **criptografar** e **autenticar** suas comunicações.

Este processo de renegociação de chaves garante que mesmo se uma ch**ave secreta partilhada** for **comprometida**, o dano será **limitado**, já que a chave será **substituída** por uma nova em um **curto período**. Além disso, o comprometimento de uma única chave **não comprometerá as comunicações futuras**, já que cada chave é independente das anteriores.

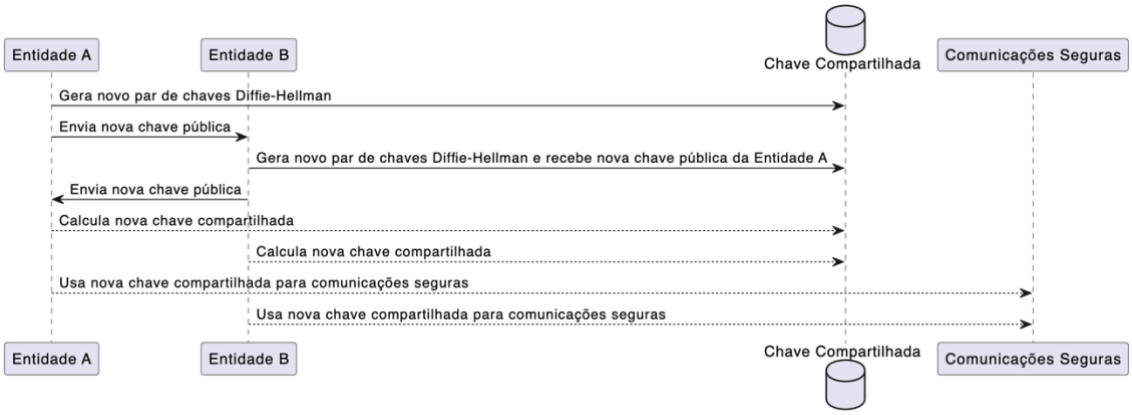

Figura 5. Diagrama de Renegociação de Chaves entre Entidades.

3. Implementação de um Protocolo Seguro

A **implementação** de um **protocolo seguro** envolve a aplicação prática de vários **algoritmos** e **técnicas** de criptografia para garantir a **confidencialidade**, **integridade** e **autenticidade** das comunicações. Neste processo, são usadas várias ferramentas e bibliotecas de criptografia, como ***OpenSSL***, que oferece uma variedade de funções criptográficas.

A implementação é um processo detalhado que requer uma compreensão sólida de **conceitos de segurança** e **criptografia**. Inclui várias etapas, como o **registo de entidades**, **autenticação**, **troca de chaves**, **troca segura de mensagens** e **renegociação de chaves**.

3.1. Registo de uma Entidade

O **registo de entidade** é a primeira etapa na implementação do protocolo de segurança. É aqui que uma entidade se torna parte do sistema de comunicação segura.

Cada entidade gera um **par de chaves RSA** usando a ferramenta ***OpenSSL***. O par de chaves consiste em uma **chave privada**, que é mantida em segredo pela entidade, e uma **chave pública**, que é enviada ao ***Gateway*** para registo. O comando OpenSSL para fazer isso é:

```
openssl genpkey -algorithm RSA -out private_key.pem
```

A chave pública gerada é então enviada ao *Gateway*. O *Gateway* mantém um registo de todas as **chaves públicas** das entidades registadas. Este registo é usado durante o processo de **autenticação** para verificar a **identidade das entidades**.

Este processo de registo de entidade ajuda a estabelecer a **identidade das entidades** no sistema. Com o **par de chaves gerado**, a entidade está pronta para participar do sistema de **comunicação segura**, onde pode **autenticar** sua identidade e estabelecer **chaves partilhadas seguras** para comunicação.

A segurança deste processo depende da segurança da **chave privada**. Se a chave privada de uma entidade for comprometida, um invasor pode se passar por essa entidade. Portanto, é essencial que as **chaves privadas sejam protegidas** com forte segurança.

3.2. Autenticação

A **autenticação** é uma etapa essencial no protocolo de segurança, pois verifica a **identidade** das partes envolvidas na comunicação. Ela usa **assinaturas digitais**, um método que combina a **chave privada** de uma entidade com um **algoritmo de hash** para criar uma **assinatura única**.

A **entidade A**, que inicia a comunicação, primeiro cria uma **assinatura digital** do conteúdo que deseja enviar. Isso é feito utilizando sua **chave privada** e um algo**ritmo de hash**, como ***SHA256***. A **assinatura digital** é criada com o seguinte comando *OpenSSL*:

```
openssl dgst -sha256 -sign private_key.pem -out signature.bin data.txt
```

Neste comando, *private_key.pem* é a **chave privada** da **entidade A**, *signature.bin* é o arquivo onde a **assinatura digital** será salva e *data.txt* é o arquivo que contém os **dados** a serem enviados.

Depois de criar a **assinatura**, a entidade A envia os **dados** e a **assinatura** para a entidade B. A entidade B usa a **chave pública** da entidade A, que foi partilhada durante o processo de registo da entidade, para verificar a assinatura. Se a assinatura for verificada com **sucesso**, isso confirma que os dados foram realmente enviados pela entidade A e não foram alterados no caminho.

O comando *OpenSSL* para **verificar a assinatura** é:

```
openssl dgst -sha256 -verify A_public_key.pem -signature
signature.bin data.txt
```

Se a verificação for bem-sucedida, a entidade B pode ter certeza de que os dados são **autênticos** e foram enviados pela entidade A.

Este processo é repetido para todas as comunicações entre as duas entidades, garantindo que as mensagens sejam sempre **autenticadas** e **seguras**. É importante ressaltar que a **chave privada** deve ser mantida em **segredo** pela entidade que a possui, pois, qualquer pessoa com acesso a esta pode se fazer passar pela entidade.

A **autenticação** é uma etapa crucial no estabelecimento de uma comunicação segura, pois ajuda a prevenir ataques de **personificação**, onde um invasor poderia tentar se passar por uma entidade para obter acesso **não autorizado** a informações ou sistemas.

3.3. Troca de Chaves

A **troca de chaves** é um componente fundamental em muitos **protocolos seguros**, permitindo que duas entidades, mesmo em uma rede insegura, estabeleçam uma **chave secreta partilhada** que podem usar para **criptografar** e **descriptografar** mensagens. O método ***Diffie-Hellman*** é um dos métodos mais usuais para a troca de chaves.
A implementação começa com cada entidade a gerar as suas **próprias chaves privadas** *Diffie-Hellman*.

No *OpenSSL*, isso pode ser feito com o seguinte comando:

```
openssl genpkey -algorithm DH -out privateKey.pem
```

Este comando gera a **chave privada** *Diffie-Hellman* e a armazena no arquivo **privateKey.pem**. A entidade B repete este processo para gerar a sua própria chave privada.
Em seguida, as entidades trocam suas **chaves públicas** *Diffie-Hellman* entre si. Isso pode ser feito por meio de um **canal de comunicação inseguro (no entanto aconselha-se sempre um seguro como TLS)**, pois mesmo que um invasor obtenha essas chaves, este não poderá obter a **chave secreta partilhada** sem a **chave privada** *Diffie-Hellman* correspondente.
Uma vez que as **chaves públicas** *Diffie-Hellman* são trocadas, cada entidade pode calcular a **chave secreta partilhada**. Eles fazem isso utilizando a sua própria **chave**

privada *Diffie-Hellman* e a **chave pública** *Diffie-Hellman* da **outra entidade**. O cálculo resultará na mesma **chave secreta partilhada** em **ambas as entidades**.
Agora, ambas as entidades têm a mesma **chave secreta partilhada**, que pode ser usada para criptografar e descriptografar mensagens. Importante, pois mesmo que um invasor tenha acesso às chaves públicas *Diffie-Hellman* e às mensagens criptografadas, sem as chaves privadas *Diffie-Hellman*, é inviável obter a chave secreta partilhada e descriptografar as mensagens.

No entanto, é importante destacar que o protocolo *Diffie-Hellman* em si **não protege contra os ataques** de ***MITM (Man-In-The-Middle)***. Para mitigar esse risco, a **autenticação** das **chaves públicas** é **necessária**, que geralmente é realizada através de **assinaturas digitais**.

3.4. Troca de Mensagens Seguras

A **troca segura de mensagens** é o cerne de qualquer **protocolo de segurança**. Isso envolve o uso da **chave secreta partilhada**, obtida por meio do processo de **troca de chaves**, para criptografar e descriptografar mensagens. Para garantir a **confidencialidade** e a **integridade** da mensagem, a criptografia e a **Message Authentication Code (MAC)** são usadas.
Nesta implementação, a **criptografia simétrica** é usada para proteger o conteúdo da mensagem. Um algoritmo de criptografia simétrica comum é o **AES (Advanced Encryption Standard)**, que oferece um **alto nível de segurança**.
A entidade que envia a mensagem, primeiro criptografa a mensagem através da **chave secreta partilhada e o AES**.

Isso pode ser feito usando o comando *OpenSSL*:

```
openssl enc -aes-256-cbc -in plaintext.txt -out
ciphertext.bin -k shared_key
```

O ficheiro **plaintext.txt** é o arquivo que contém a mensagem original, **ciphertext.bin** é o arquivo onde a mensagem criptografada será salva e **shared_key** é a chave secreta partilhada.

Depois de criptografada, a mensagem pode ser enviada para a outra entidade. Mesmo se um invasor intercetar essa mensagem, ele não conseguirá descodificar o conteúdo sem a **chave secreta partilhada**.
Para garantir a **integridade** da mensagem, um **MAC (Message Authentication Code)** é gerado. O MAC é um resumo criptográfico da mensagem, criado usando um algoritmo **HMAC (Hash-based Message Authentication Code)** e a chave secreta partilhada.

O *OpenSSL* pode ser usado para gerar o MAC:

```
openssl dgst -sha256 -hmac shared_key -out mac.bin
ciphertext.bin
```

O **MAC** é então enviado junto com a mensagem criptografada. Quando a mensagem é recebida, o **MAC é verificado**, comparando-o com um novo MAC gerado a partir da mensagem recebida. Se eles corresponderem, a **integridade da mensagem é confirmada**.

Esses passos garantem que as mensagens sejam **trocadas de maneira segura e confiável**, protegendo a **confidencialidade** e a **integridade** da comunicação.

3.5. Renegociação de Chaves

Renegociar chaves em intervalos regulares é uma prática de segurança recomendada para garantir a segurança contínua da comunicação. Isso ocorre porque, com o tempo e o uso contínuo da mesma chave, pode haver um **aumento no risco** de a **chave ser descoberta** ou **comprometida**.
A renegociação de chaves envolve a **criação** e **troca** de uma nova **chave secreta partilhada,** semelhante ao processo inicial de troca de chaves.

No *OpenSSL*, isso pode ser feito usando os comandos:

```
openssl genpkey -algorithm DH -out new_privateKey.pem
```

Esse comando gera um novo conjunto de parâmetros *Diffie-Hellman* e os armazena no arquivo **new_privateKey.pem**. A entidade B repetirá esse processo para gerar a sua própria **chave privada**.

Depois de geradas, as novas **chaves públicas** *Diffie-Hellman* são trocadas entre as entidades. Cada entidade usa a nova **chave pública** *Diffie-Hellman* da outra entidade e sua própria nova **chave privada** *Diffie-Hellman* para gerar a nova **chave secreta partilhada**.

Com a **nova chave secreta partilhada** em vigor, todas as futuras comunicações serão criptografadas usando essa nova chave. Isso ajuda a manter a segurança das comunicações, pois mesmo que um invasor tenha obtido a chave anterior, ele não será capaz de **decifrar novas mensagens**.

O processo de **renegociação de chaves** deve ocorrer em **intervalos regulares** para garantir a **segurança contínua**. O tempo exato entre as renegociações de chaves pode variar dependendo de vários fatores, incluindo a **quantidade de dados transmitidos** e os requisitos de **segurança específicos**.

Em resumo, a **renegociação de chaves** é uma prática importante para garantir a **segurança** de um **protocolo de comunicação**. É uma forma de garantir que, mesmo que uma chave seja **comprometida**, o impacto seja **limitado** e a **comunicação** segura possa continuar.

4. Plano de Resposta a Incidentes

Apesar das medidas **preventivas**, **violações de segurança** ainda podem ocorrer. Nesses casos, um **plano de resposta a incidentes** deve ser ativado. O primeiro passo é **identificar** e **conter** o incidente para **minimizar o dano**. Em seguida, a ameaça deve ser **erradicada** e as **operações devem ser recuperadas**. Após a resolução do incidente, uma **análise pós-incidente** deve ser realizada para aprender com o evento e melhorar as **medidas de segurança** futuras.

5. Conclusão

Baseado na informação fornecida, é possível concluir que este documento apresenta o **projeto** e a **implementação** de um **protocolo de segurança criptográfica** destinado a sistemas de **chat seguro**. O protocolo foi desenvolvido para permitir **comunicação segura** e **autenticada** através de um *Gateway* central, com uma arquitetura **descentralizada** e privada.

O sistema proposto aplica os **princípios fundamentais** da **criptografia** e da **segurança das redes**, assegurando **autenticação**, **confidencialidade**, **integridade** e **disponibilidade** nas comunicações. O protocolo inclui várias etapas: **registo da entidade**, **autenticação**, **troca de chaves**, **troca segura de mensagens** e **renegociação de chaves**.

A implementação do protocolo faz uso de várias ferramentas e bibliotecas de criptografia, como *OpenSSL*. O processo de implementação requer uma compreensão sólida de conceitos de **segurança** e **criptografia**, e é acompanhado por medidas rigorosas para proteger a **chave privada** do *Gateway* central e as **chaves privadas** das entidades envolvidas.

Em resumo, o protocolo proposto oferece uma **solução robusta** e **segura** para a **comunicação em sistemas de chat**, reduzindo a dependência de um **único ponto de falha** e aumentando a **resiliência do sistema**.

Embora redundante e repetitivo vários conceitos ao longo do projeto considerámos importante para melhor compreensão do mesmo.

6. Bibliografia

1. Stallings, W. (2006). Cryptography and Network Security: Principles and Practices. Pearson Education.

2. Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE transactions on Information Theory, 22(6), 644-654.

3. Rescorla, E. (2001). SSL and TLS: Designing and Building Secure Systems. Addison-Wesley Professional.

4. Menezes, A., Van Oorschot, P., & Vanstone, S. (1996). Handbook of Applied Cryptography. CRC Press.

5. Ferguson, N., & Schneier, B. (2003). Practical Cryptography. Wiley Publishing, Inc.

6. OpenSSL Project. (2022). OpenSSL Cryptography and SSL/TLS Toolkit. https://www.openssl.org/

7. National Institute of Standards and Technology (NIST). (2001). Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197. https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

8. Bellare, M., Canetti, R., & Krawczyk, H. (1996). Keying hash functions for message authentication. In Annual International Cryptology Conference (pp. 1-15). Springer, Berlin, Heidelberg.

9. Kohnfelder, L. (1978). Towards a practical public-key cryptosystem. MIT.

10. Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc.

11. Slides das aulas

12. Lab. Criptography Enunciado