Segurança e Privacidade

TP1 - Partilha de dados com encriptação

Eduardo Neves - 2018281324
Frederico Cardoso - 2012138904

Universidade de Coimbra

16 de outubro de 2022

Introdução

Nos tempos que correm, virtualmente toda a partilha de informação, é feita através de canais de comunicação virtuais, através da Internet. Tal leva a que certas medidas tenham que ser adoptadas para garantir a **integridade**, a **confidencialidade** e a **autenticidade** dos dados partilhados. Uma forma de garantir estes três aspectos é recorrendo à encriptação da informação. Pretendeu-se, com este trabalho prático, que fosse implementada uma simples, mas segura, comunicação com transmissão de informação sensível entre duas entidades fictícias, de forma a explorar como podem ser aplicadas estas medidas, que protocolos e algoritmos devem ser utilizados e que garantias são devolvidas. Para tal, uma troca de chaves deve ser feita entre estas entidades Explorou-se também o caso de haver entidades externas à escuta do canal, para uma demonstração prática de que os dados partilhados apenas conseguem ser lidos por quem lhes diz respeito.

1 Troca de chaves Diffie-Hellman

O algoritmo Diffie-Hellman (**DH**) é um dos métodos mais utilizados para a geração e troca de chaves através de canais considerados não seguros. O seu modelo matemático de exponenciação de módulos, baseado na geração de chaves públicas e privadas para cada utilizador do canal, consiste na criação de chaves simétricas comuns a dois utilizadores. Esta chave simétrica surge quando as entidades trocam entre si as suas chaves públicas e, obtendo a chave pública de outra entidade, juntando a sua própria chave privada. Esta chave é igual nas duas extremidades e, dado o modelo matemático usado e como não foi partilhada nenhuma informação sobre as chaves privadas pelo canal, torna a sua descodificação bastante complexa, espacial e temporalmente. Esta garantia de segurança é o primeiro passo para uma partilha segura de informação, posteriormente, porque a chave simétrica, sendo comum às duas entidades e secreta a todas as outras, será utilizada em algoritmos de encriptação e desencriptação de ficheiros, como é o caso do **AES**, que será referenciado a seguir. A única desvantagem provém da falta de autenticidade das mensagens, facto que pode ser facilmente atenuado pelo algoritmo de encriptação AES, fruto do modelo de operação GCM.

2 Andvanced Encryption Standart

O Andvanced Encryption Standart (**AES**) é um modelo de encriptação de bloco simétrica muito utilizado em segurança informática. A possibilidade de utilização de chaves até 265 bits permite uma combinação bastante mais criteriosa, o que aumenta exponencialmente o grau de complexidade de desencriptação por parte dos atacantes. Havendo uma garantia de confidencialidade das chaves simétricas utilizadas, como é o caso do algoritmo DH, o AES finda ser a camada final da troca segura de informação entre duas entidades por um canal não privado. A par do DH, utilizando o modo de operação de Galois/Counter (**GCM**), que permite uma *assinatura* da entidade que envia os dados, garante-se a autenticidade e a integridade da informação. Para além disto, é um método que pauta pela sua performance bastante superior a outros modelos do AES.

3 Arquitetura

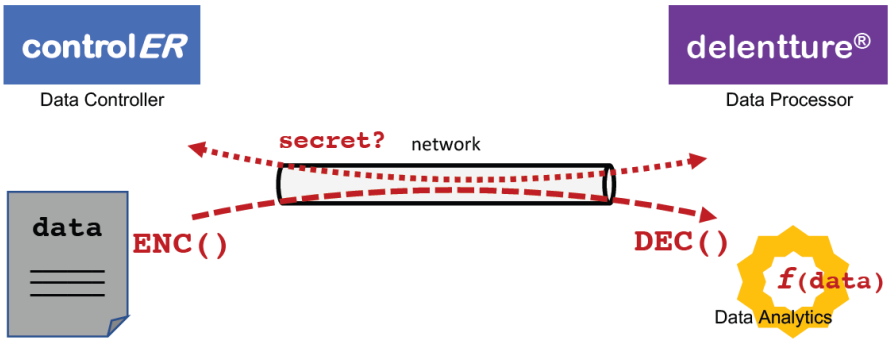

Figura 1: Modelo de comunicação e partilha *(Fonte: Enunciado)*

A controlER (entidade A) é uma empresa fictícia que presta serviços de empréstimos aos consumidores. Recentemente, a empresa notou aumentos significativos em violações contratuais. Como a empresa A não possui nenhuma equipa de analistas/cientistas de dados, decidiu contratar a empresa delentture (entidade B), uma empresa de consultoria com uma enorme especialização em ciência de dados, para obter informações relevantes em estratégias de restrição de aprovações contratuais. Dada a sensibilidade da informação a ser partilhada por um canal de comunicação público, a controlER quer assegurar que nenhuma desta é extraviada e analisada por entidades externas.

Para resolver o problema, em primeiro lugar, aberto o canal de comunicação, a entidade A tem que acordar com a entidade B a troca das respetivas chaves públicas. Tendo cada uma dessas entidades as chaves da outra, é gerada, localmente, uma chave simétrica partilhada, utilizando o algoritmo DH. A entidade A utiliza, então, o protocolo AES-GCM, com a sua chave simétrica, para encriptação da sua informação confidencial, para poder transmitir à entidade B, que, também contendo a chave simétrica, a consegue revelar. Utilizando este algoritmo a autenticidade e integridade dos resultados é garantida. Após a desencriptação, a entidade B utiliza os métodos de análise de dados que ache relevantes, criando um relatório com os resultados para ser transmitido à entidade A. Esta última transmissão é também efetuada após a encriptação dos dados utilizando o mesmo algoritmo e a mesma chave simétrica. Por fim, após a transmissão, a entidade A desencripta o relatório final para poder analisar e aplicar as medidas que ache mais convenientes.

4 Implementação

Para a implementação prática desta arquitetura anteriormente descrita, recorreu-se principalmente à biblioteca *criptography* [1] do Python, de onde foram extraídas funções referentes aos algoritmos DH [2], AES-GCM [3] e HKDF, para resolução de problemas de derivação de chaves através de *hashing*. A abordagem seguida foi a seguinte:

1. **Geração de parâmetros para o DH.** Estes foram utilizados para criar as chaves pública e privada de A;
2. **Envio da chave pública de A para o canal e recepção por B.**
3. **Extração de parâmetros para o DH.** A entidade B, através da chave pública de A, extrai os parâmetros que foram usados para aplicar o DH, como forma de *aperto de mão* entre as entidades. Através destes, gera as suas próprias chaves pública e privada;
4. **Envio da chave pública de B para o canal e recepção por B.**
5. **Geração da chave simétrica partilhada.** Tendo cada entidade a chave pública da outra, gerou-se em ambas as extremidades a chave a utilizar para encriptação/desencriptação;
6. **Hashing - geração da *chave secreta* e do *nonce*.** Através do algoritmo HKDF gerou-se um par de valores, a serem utilizados na encriptação do ficheiro, uma *chave secreta*, resultado

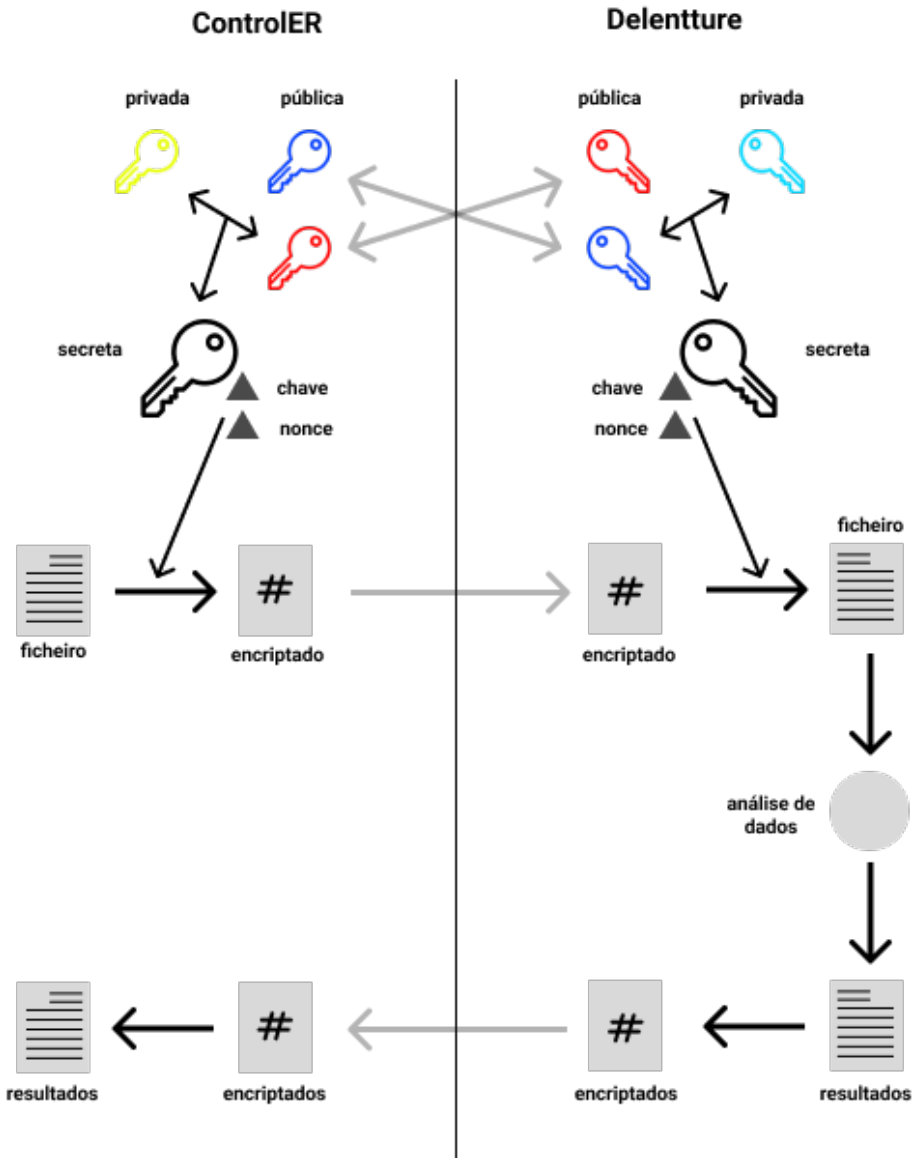

Figura 2: Modelo de implementação dos algoritmos de encriptação

da derivação da chave simétrica para 32-bytes e um *nonce*, resultado da derivação da chave simétrica para 12-bytes. Esta derivação é feita de igual forma dos dois lados, através de um segundo *aperto de mão*, gerando, portanto, pares de valores simétricos.

7. **Encriptação dos ficheiros de informação por A e envio pelo canal para B.**
8. **Recepção dos ficheiros de informação e desencriptação.** Esta desencriptação é feita através do mesmo par de valores gerado por A localmente, visto que são simétricos.
9. **Análise dos dados e elaboração do documento final.** São feitas duas operações de análise de dados, sendo a primeira uma visualização geral em vários histogramas sobre as violações contratuais verificadas, e a segunda uma categorização por faixa etária dos clientes da empresa. É criado um documento final em formato *pdf* para envio à entidade A.
10. **Encriptação do documento final por B e envio pelo canal para A.**
11. **Recepção do documento final e desencriptação.**

5 Atacantes

Como se pode depreender, uma entidade à escuta só terá acesso às quatro parcelas partilhadas: as chaves públicas de A e de B, a informação sensível encriptada e o ficheiro de resultados encriptado. Não tendo acesso às chaves privadas dos utilizadores, não consegue gerar a chave simétrica necessária para descodificar os ficheiros encriptados. Para além disso, se por algum acaso tivesse acesso à chave simétrica, precisaria ainda da informação sobre o *aperto de mão* utilizado para gerar a chave secreta e o *nonce* utilizados, pois é com estes que é feita a encriptação.

6 Resultados

A nível de complexidade espacial, no caso do DH é reduzida, visto que uma chave apenas ocupa umas centenas de bytes de memória. No AES-GCM a complexidade espacial é linear, dependendo apenas do tamanho do ficheiro. A complexidade temporal é bastante reduzida no algoritmo de encriptação, no entanto a geração dos parâmetros utilizados para a geração das chaves pode ser bastante superior. A nível de segurança da comunicação e partilha, dada a boa implementação dos algoritmos, atingiram-se os níveis de autenticidade e integridade esperados.

Conclusão

A utilização do modelo DH, padrão desde os anos 70, dada a sua enorme complexidade de descodificação, prova-se mais uma vez, bastante eficaz. Mesmo num simples programa como o desenvolvido, seriam necessários anos de computação para poder descobrir a chave secreta utilizada pelas duas entidades, o que permitiu a integridade da informação partilhada, tendo-lhe ainda sido acrescentada a encriptação AES-GCM, complementando o programa com a autenticidade requerida. Quanto à análise dos dados, descobriu-se que, em média, as violações ocorridas nos contratos acabavam por beneficiar os indivíduos, como é o caso de alguns cancelamentos de contratos altos. Podemos também deparar que a maioria das violações ocorriam nas faixas etárias dos 40 anos em diante.

Referências

[1] "Cryptography.io," https://cryptography.io/en/latest/.

[2] "Dh library," https://cryptography.io/en/latest/hazmat/primitives/asymmetric/dh/.

[3] "Aes library," https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html.