universidade de aveiro  deti departamento de electrónica, telecomunicações e informática

Projects 2+3: Digital Rights Management

Licenciatura em Engenharia Informática
Segurança Informática e nas Organizações
Docente: João Barraca

Alexandre Rodrigues, 92951
Gonçalo Matos, 92972

Ano letivo 2020/2021

Índice

Introdução

Proposto como segundo e terceiro trabalho prático na cadeira de Segurança Informática e nas Organizações, este projeto teve como objetivo o desenvolvimento de um sistema de comunicações seguras entre cliente e servidor com o objetivo da distribuição de conteúdos multimédia.

Ao longo deste relatório serão explicadas as metodologias adotadas e o fluxo de trabalho ao longo do projeto assim como um diagrama de sequência para melhor retratar as comunicações entre o cliente e o servidor.

Protocolo implementado

Para o desenvolvimento deste projeto criámos um protocolo de comunicação segura entre o servidor e os seus potenciais clientes.

Na nossa implementação assumimos que vários clientes podem estar conectados ao servidor em simultâneo, sendo atribuído a cada um uma sessão. Por sua vez, em cada cliente, pode haver um e apenas um utilizador autenticado, podendo no entanto vários utilizadores fazerem uso da mesma sessão para o consumo de multimédia, desde que em momentos distintos.

Asseguramos a confidencialidade e integridade de todas as comunicações que o justificam e a autenticação e isolamento dos intervenientes, através de vários mecanismos.

Abaixo descrevemos os procedimentos que definimos para a comunicação, pela ordem temporal em que estes ocorrem.

Autenticação mútua entre o cliente e o servidor

Em todas as comunicações descritas ao longo deste protocolo há uma autenticação mútua entre o cliente e o servidor. Para tal, criámos uma entidade certificadora, cuja certificado armazenámos na pasta */certsca*, que é assumida pelo sistema como de confiança e a partir da qual gerámos um certificado para o cliente e outro para o servidor, localizados, respetivamente, em */certificates/client_localhost.crt* e */certificates/server_localhost.crt*.

Para cada pedido do cliente, seja GET ou POST este envia o seu certificado no cabeçalho, assim como uma assinatura com a sua chave privada do conteúdo enviado (seja este cifrado ou não).

O servidor procede na mesma forma, enviando o seu certificado e a assinatura da resposta nos cabeçalhos da resposta aos pedidos do cliente.

Caso a assinatura não seja considerada válida, o servidor descarta o pedido e o cliente a resposta.

Proteção dos ficheiros no servidor

Todo o conteúdo do servidor está cifrado com recurso à chave guardada no ficheiro */server/key.txt*, nomeadamente a base de dados de utilizadores em */server/licenses.json* e todos os ficheiros multimédia na pasta */server/catalog*.

Inicialização do servidor

Quando o servidor é inicializado, são executadas várias tarefas:

1. Carregamento do grupo de parâmetros para a geração das chaves do algoritmo *Diffie-Hellman*, que está armazenada no ficheiro *server/parameters*;
2. Carregamento e desencriptação do catálogo de ficheiros multimédia a disponibilizar, que estão encriptados em disco, na pasta */server/catalog*;
3. Carregamento da chave privada do certificado do servidor, que está armazenada no ficheiro *keys/server_localhost.pem*;
4. Carregamento do certificado do servidor, que está armazenado em *certificates/server_localhost.crt;*
5. Inicialização do dicionário de sessões;
6. Disponibilização do servidor na porta local 8080.

Com esta inicialização são disponibilizados vários *endpoints*, que são utilizados pelo cliente para realizar as várias operações necessárias ao consumo dos conteúdos.

Inicialização do cliente

A inicialização do cliente consiste em:

1. Carregamento da chave privada do certificado, que está armazenada no ficheiro *keys/client_localhost.pem*;
2. Carregamento do certificado, que está armazenado em *certificates/client_localhost.crt;*
3. Consulta do grupo de parâmetros para a geração das chaves do algoritmo *Diffie-Hellman*, através de um pedido GET desencriptado para */api/parameters*;
4. Com base nestes parâmetros, é gerado o par de chaves privada e pública do cliente para o algoritmo *Diffie-Hellman*.

Negociação do pacote de cifras

Uma vez inicializado, o cliente consulta os pacotes de cifra disponíveis para a comunicação com o servidor através de um pedido GET desencriptado para */api/protocols*, do qual recebe uma **lista de conjugações de algoritmos de cifra e de síntese e modos de cifra**.

Os pacotes que implementámos foram:

Algoritmo de cifra	**Modo de cifra**	**Algoritmo de síntese**
AES	CBC	SHA512
AES	OFB	SHA512
3DES	CBC	BLAKE2
3DES	OFB	BLAKE2

Dos vários pacotes disponíveis, é escolhido um pelo utilizador, que é guardado.

Registo da sessão no servidor e negociação das chaves de sessão

Para negociar as chaves de sessão, o cliente faz POST desencriptado da sua **chave pública** e **do pacote de cifras** escolhido no passo anterior pelo utilizador para */api/session*.

Ao receber este pedido, o servidor vai gerar um ID para a sessão e um par de chaves privada e pública para a mesma. Com a chave privada gerada e a chave pública recebida do cliente, vai gerar a chave de sessão, utilizada mais tarde para a cifra das comunicações. Estes valores serão armazenados no dicionário de sessões, em conjunto com o pacote de cifras enviado pelo cliente.

Terminado todo este processamento, é enviada uma resposta desencriptada ao cliente com a chave pública do servidor e com o ID da sessão no cabeçalho.

Com esta resposta, o cliente guarda o ID da sessão recebido, que irá utilizar para se identificar em todos os contactos seguintes e gera ainda a chave de sessão a partir da chave privada e a chave pública do servidor recebida.

Encriptação e validação da integridade das comunicações

A partir deste momento o cliente encontra-se registado no servidor com um ID de sessão e as comunicações passam a ser encriptadas com recurso ao pacote de cifras negociado.

Caso o ID de sessão não seja enviado no cabeçalho dos pedidos para os endereços descritos abaixo, terão uma resposta de erro, uma vez que esta é fundamental para identificar a chave de sessão utilizada para encriptar as respostas e para a geração dos códigos de integridade.

Em cada pedido, são ainda enviados os códigos MIC e MAC. O MIC para garantir que não há erros na comunicação que consiste numa síntese do *payload* do pedido e o MAC, que permite a deteção de alterações intencionais às mensagens trocadas, que consiste na síntese do *payload* do pedido concatenado com a chave de sessão.

Registo com *hardware tokens*

Apesar de ter atribuído um ID de sessão, neste momento, o cliente ainda não tem acesso aos conteúdos multimédia disponibilizados pelo servidor. Para tal tem de se autenticar.

Começamos por descrever o processo de registo. Quando selecionado pelo utilizador, é-lhe pedido o nome de utilizador e a palavra-passe que pretende utilizar na sua conta. De seguida, é criada uma assinatura para a concatenação do nome de utilizador com a palavra-passe definida com recurso à chave privada do Certificado de Autenticação do Cartão de Cidadão, sendo-lhe pedido o PIN para este efeito.

Segue-se o envio de um pedido POST ao servidor com o par de dados de autenticação, a assinatura gerada e ainda o Certificado de Autenticação e os certificados intermédios do Cartão de Cidadão, para o URL */api/newuser*.

Ao receber este pedido, o servidor valida a cadeia de certificação do certificado recebido e a assinatura da concatenação do nome de utilizador e palavra-passe recebidos. De seguida, carrega e desencripta o conteúdo do ficheiro */server/licenses.json* para validar que o nome de utilizador a registar não existe ainda na base de dados.

Caso todas as validações anteriores tenham sido bem sucedidas, o utilizador é criado, é gerada uma síntese da palavra-passe recebida para cada algoritmo de cifra suportado pelo servidor e é-lhe atribuída uma licença por defeito, que permite 3 visualizações até um limite de 5 minutos seguintes àquele momento.

O nome de utilizador, as sínteses da palavra-passe geradas, a licença e o Certificado de Autenticação são guardados no ficheiro cifrado */server/licenses.json*, que é agora atualizado.

Autenticação com *hardware tokens*

Para se autenticar, a interação com o utilizador é bastante semelhante, sendo-lhe pedidos os dados de autenticação. No entanto, o tratamento destes difere, a começar pela criação de uma síntese da palavra-passe inserida, que é utilizada para a geração da assinatura com recurso ao Cartão de Cidadão, de forma análoga ao registo, mas agora com a concatenação do nome de utilizador com síntese da palavra-passe em vez da palavra-passe em claro.

Segue-se um pedido POST ao servidor para o URL */api/auth* com o envio do par de dados de autenticação e da assinatura. Ao recebê-lo, o servidor valida a assinatura recebida como certificado obtido no registo e em caso de sucesso e da validação dos dados de autenticação, a sessão passa a estar autenticada e é associada ao utilizador.

Consumo de conteúdos multimédia

O acesso à listagem dos conteúdos multimédia é feito através de pedidos GET e POST para os endereços */api/list* e */api/download*, respetivamente. Nestes pedidos, há uma validação adicional à sessão, que para além de ter de ser identificada pelo cliente, tem de estar autenticada por um utilizador, caso contrário é respondida uma mensagem de erro.

No *download*, o ID do conteúdo e número do *chunk* a obter são enviados no *payload* encriptado, pelo que não é possível alguém que intersete as comunicações identificar o que o cliente está a consumir.

Para garantir uma melhor qualidade de reprodução, caso a resposta a um pedido de *download* não tenha um MIC ou MAC válido, o pedido é repetido até 4 vezes, até que os códigos de integridade seja válidos, caso contrário, o *chunk* é ignorado.

Rotação das chaves com base no chunk

Para aumentar a robustez das comunicações, implementámos um mecanismo que para cada *chunk* enviado pelo servidor, faz *append* do inteiro em *bytes* à chave partilhada antes de fazer o processo de cifra. O cliente faz o mesmo procedimento para a decifra. Assim, para cada pedaço de média trocado, será utilizada uma chave diferente no processo de cifra.

Gestão das licenças

Para permitir a gestão das licenças de consumo de conteúdo, criámos um endereços */api/license* e */api/renew*, que recebem, respetivamente pedidos GET e POST para a consulta da licença atual e para a sua renovação. Ambos estão restritos a utilizadores autenticados.

O pedido de renovação é sempre aceite pelo servidor, sendo a renovação feita para 3 visualizações para 3 visualizações até um limite de 5 minutos seguintes àquele momento.

Log out

Para permitir a utilização do cliente por vários utilizadores na mesma sessão, é possível o *log out* do utilizador através de um pedido POST para o endereço */api/auth* com o corpo logout:True. Este *endpoint* também está limitado a utilizadores autenticados.

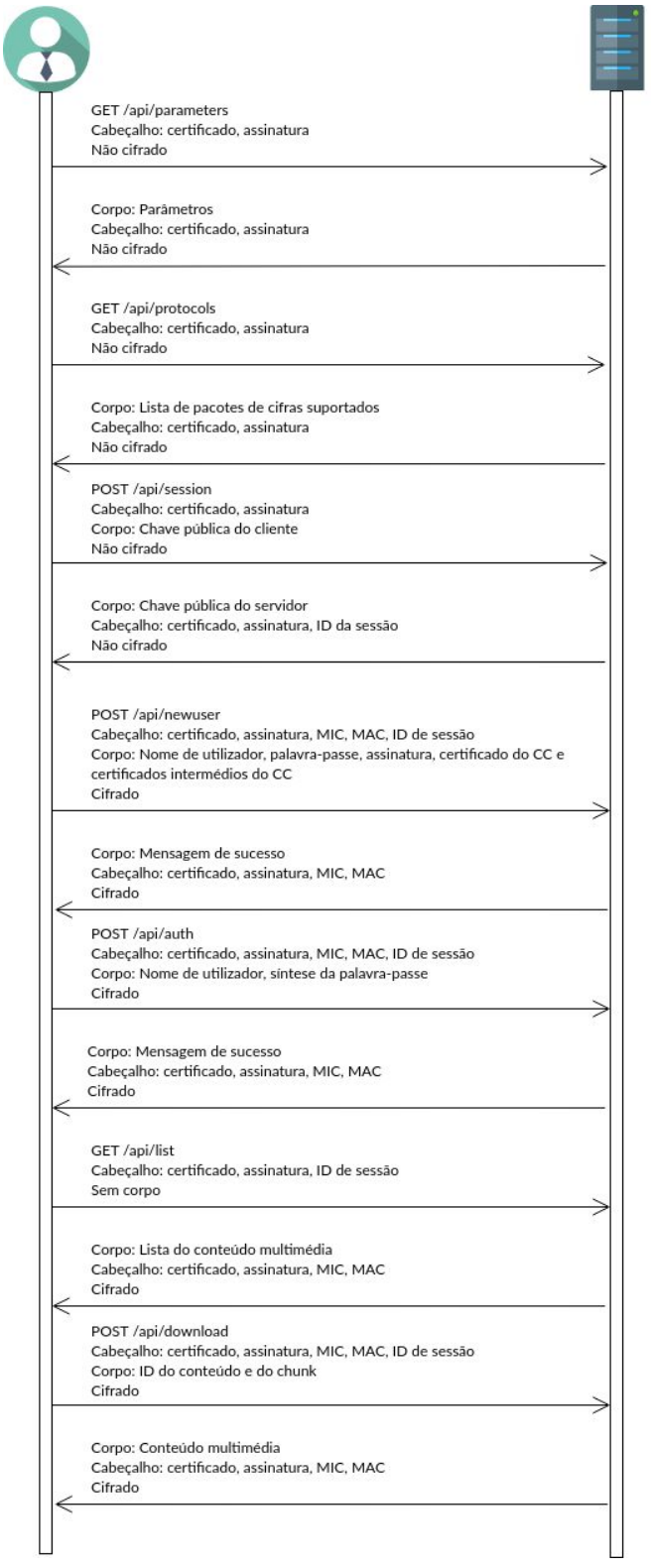
Gestão das sessões

Antes de terminar a execução e para prevenir eventuais utilizações indevidas do ID da sessão por atacantes, o cliente envia um pedido POST para o endereço */api/sessionend*, sendo esta eliminada do dicionário de sessões ativas do servidor. O acesso a este *endpoint* não requer que o utilizador esteja autenticado, mas apenas uma sessão válida.

Mesmo que não seja terminada pelo cliente por algum erro ou interrupção do seu programa, cada sessão tem associado um prazo de 2 horas, após o qual é considerada inválida pelo servidor e deixa de ser aceite em pedidos.

Diagrama de Sequência

Neste diagrama ilustramos o fluxo de comunicações principais entre o cliente (à esquerda) e o servidor (à direita). Estas incluem todos os pedidos do cliente desde a sua inicialização, passando pela negociação das cifras e da chave de sessão, pelo registo, autenticação e por fim consumo de um conteúdo multimédia.



Conclusão

Com a realização deste trabalho prático foi possível aprofundar os nossos conhecimentos sobre o desenvolvimento de um sistema de comunicações seguras entre cliente e servidor, através de autenticação dos clientes usando o cartão de cidadão e autenticação do servidor usando certificados X.509, com o objetivo da distribuição de conteúdos multimédia.

Foram realizados todos os pontos propostos no guião do projeto. Alguns pontos suscitaram algumas dúvidas, contudo com a disponibilidade, que tanto somos gratos, do Prof. João Barraca que sempre esteve disponível para esclarecer as nossas dúvidas foi possível que fossem concluídos.

Referências

Protocolo de autenticação
https://aaronparecki.com/oauth-2-simplified/

Mecanismos de criptografia e autenticação
https://cryptography.io/en/latest/

Gestão de sessões
https://blog.sean-wright.com/session-management-client-side-vs-server-side/
https://www.packetlabs.net/session-management/
https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

Slides teóricos da UC, do professor João Paulo Barraca

Guiões práticos da UC, do professor João Paulo Barraca