

Segurança de Redes e Sistemas de Computadores

2018/2019, 2º Sem.

Ficha de Implementação do Trabalho Prático nº 1

Secure Real Time Media Streaming System

Grupo

Nº NNNNN Nome Apelido

Nº NNNNN Nome Apelido

1. Introdução e caracterização do trabalho desenvolvido

1.1 Colocar X nas colunas de acordo com a sua implementação

FASE 1: Implementação e completude do trabalho	SIM	NÃO	Parcial ou incompleto
Foram implementados totalmente todos os requisitos da FASE 1 (Fase Obrigatória)			
Com as configurações criptográficas testadas (ficheiro <i>ciphersuites.conf</i> , o trabalho funciona corretamente e permite a reprodução do <i>stream</i> (media) no VLC, com qualidade e sem cortes, estando o canal entre o <i>Stream Server</i> e o <i>Proxy</i> protegido de acordo com a especificação do protocolo para a Fase 1 (no anexo do enunciado)			
A minha implementação dos objetivos da Fase 1 implementou a noção de <i>Secure UDP Sockets</i> , suportando o canal de comunicação seguro de forma transparente ao código do <i>Stream Server</i> e do <i>Proxy</i>			
Com a minha implementação da Fase 1, o código do <i>Stream Server</i> e do <i>Proxy</i> , apenas difere do código original de cada um desses componentes em menos de 10 linhas de código			
Na minha implementação da Fase 1, utilizo as parametrizações criptográficas no ficheiro de configuração <i>ciphersuites.conf</i> , mas as chaves criptográficas ou chaves de MACs, são mantidas numa ou mais keystores (do tipo JECKS)			
O ficheiro de configuração (<i>ciphersuites.conf</i>) pode suportar mais do que uma parametrização criptográfica, de acordo com a definição de diferentes <i>endpoints</i> (IP ou IP,Porto).			
No caso de utilização de keystores para gestão das chaves, usa-se apenas uma única keystore que pode conter mais do que uma chave simétrica ou MAC, sendo estas armazenadas em diversas " <i>entries</i> " dessa única keystore			
No caso de utilização de várias keystores para gestão das chaves, cada keystore contém apenas uma chave simétrica e uma chave MAC, sendo estas armazenadas em " <i>entries</i> " dessa única keystore			

2. Aspetos de clarificação sobre a tabela do ponto 1.

Em relação às tabelas dos pontos 1 (1.1 ou 1.2), no caso de ter indicado “Parcial ou Incompleto” em alguma das linhas, pode incluir no seguinte quadro qualquer comentário de clarificação ou justificação da sua caracterização.

3. Keystore (ou Keystores) utilizadas e seu acesso para ensaio e verificação experimental

Inclua na seguinte tabela, a ou as keystores usadas, bem como as passwords de acesso para que se possa proceder à verificação e ensaio experimental. Se não usou keystores não precisa de preencher esta informação.

FASE 1:

Keystore usada pelo Proxy	Nome da Keystore	Password de Proteção
Keystore usada pelo Stream Server	Nome da Keystore	Password de Proteção

4. Exemplo de configuração *ciphersuites.conf*

Apresente um exemplo de configuração do seu ficheiro *ciphersuites.conf* tal como é usado pelo *StreamServer* e pelo *Proxy*, na implementação da Fase 1 do seu trabalho.

5. Configurações criptográficas usadas e testadas experimentalmente na comprovação do funcionamento correto da implementação

5.1 *Ciphersuites* configuráveis (ficheiro *ciphersuites.conf*) que foram testadas na implementação da FASE 1

Indique em cada uma das seguintes linhas, as *ciphersuites* que ensaiou e verificou, atendendo à sua diversidade, de modo a comprovar a generalidade da sua solução em relação à flexibilidade para suportar diferentes configurações criptográficas envolvendo algoritmos criptográficos simétricos (de bloco ou em cadeia), MACs (HMACs ou CMACs) ou alternativamente Funções de seguras Hash. Deve preencher a tabela de acordo com as *ciphersuites* configuradas que foram testadas e que comprovou experimentalmente e assim pode demonstrar se solicitado.

Ciphersuite (Identifique, por ex., AES/CTR/PKCS5Padding)	MACs Identifique as construções HMAC ou CMAC utilizadas para a Ciphersuite utilizada no protocolo da Fase 1 (ou alternativamente funções seguras de Hash que tenham também disso utilizadas)	Tamanho (em bits) da Chave Criptográfica	Tamanho (em bits) da chave do MAC (HMAC ou CLAC)

6. Outros aspetos da implementação relativos à FASE 1

6.1 Vetores de inicialização

Indicar como são gerados, os vetores de inicialização, para os modos que deles necessitam e como são os mesmos sincronizados/estabelecidos entre o Stream Server e o proxy, nomeadamente no caso de estes serem gerados e distribuídos dinamicamente

--

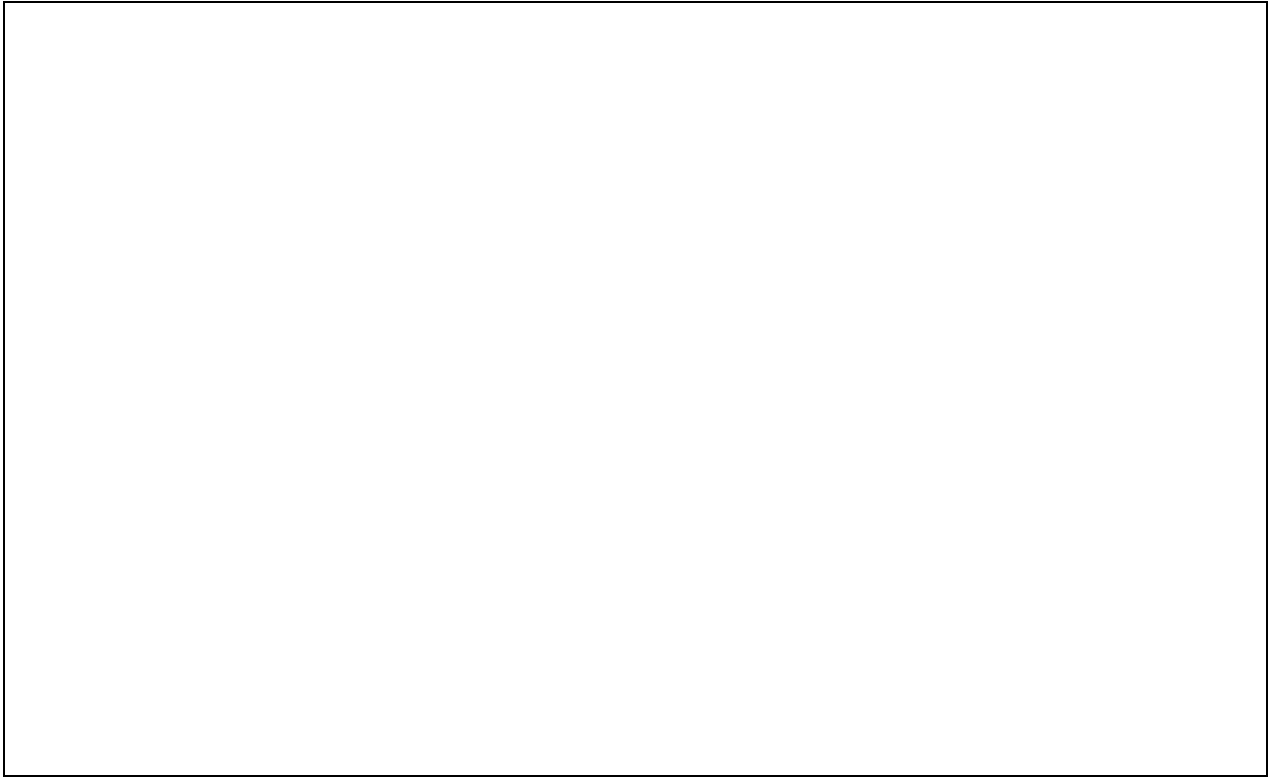
6.2 Indique se e como é que na sua solução são implementados NONCES e como é que protege e trata a possibilidade de detecção de ataques de *Message Replaying*

6.3 Indique se e como é que na sua solução são implementados N°s de Sequência e como é que protege e trata a possibilidade de detecção de ataques de *Message Replaying*

- 6.4 Indique se e como é que na sua solução são implementados TIMESTAMPS como forma de verificação de freshness dos pacotes e se tal permite detectar ataques de replaying, nomeadamente no caso do ataque ser conjugado com ataques à sincronização de relógios do StreamServer e Proxy.

- 6.5 Indique qualquer diferença do protocolo que implementou na Fase 1, comparativamente à especificação de referencia que foi indicada no enunciado (tal como inicialmente especificado no Anexo 1)

- 6.6 Inclua qualquer aspeto que considere relevante para que seja considerada na verificação e avaliação do seu trabalho, em relação à implementação da Fase 1 (obrigatória).



7 - Fase 2 . Preencher apenas no caso de ter implementado a Fase 2

Stream-DTLS protocol with dynamic configurations established by an Authentication and Key-Establishment Protocol

7.1 Implementação da Fase 2 (Preencher apenas se foi implementado)

FASE 2: Implementação e completude do trabalho	Sim	Não	Parcial ou incompleto
Desenhei e implementei uma solução para a Fase 2			
Foi implementada a Fase 2, atendendo aos requisitos iniciais, tendo sido desenhada uma solução para a integração de um servidor KDC (<i>KDC-Ticket Line</i>) que implementa a noção de serviço de autenticação para o proxy, bem como integra a noção de Movie-Coins e emissão de Tickets, suportando a configuração e estabelecimento dinâmicos dos parâmetros criptográficos que o proxy e o StreamServer usam para reprodução e emissão de streams protegidos no canal UDP.			
De acordo com a minha solução para a Fase 2, o trabalho funciona corretamente e permite a reprodução do stream (media) no VLC, com qualidade e sem cortes			
De entre os modelos e protocolos de distribuição de chaves que usam apenas criptografia simétrica estudados (aulas teóricas), qual o modelo e protocolo que usou como referencia ou inspiração para a implementação do protocolo de autenticação e estabelecimento dinâmico de <i>ciphersuites</i> , chaves e demais parâmetros criptográficos por parte do KDC (KDC-Ticket Line) para estabelecimento do canal seguro de streaming (UDP) usado entre proxy e SteramServer.	Identifique o protocolo (ex., Neuman-Stubblebine, etc) ...		

7.2 Modularidade, transparência e estruturação da implementação

FASE 2: (Não preencha se não implementou) Comparativamente ao código inicialmente distribuído para o <i>Proxy</i> , indique o número de linhas correspondentes a alterações produzidas, após a implementação da FASE 2	
FASE 2: (Não preencha se não implementou) Comparativamente ao código inicialmente distribuído para o <i>StreamServer</i> , indique o número de linhas correspondentes a alterações produzidas, após a implementação da FASE 2	
FASE 2: (Não preencha se não implementou) Identifique o algoritmo PASSWORD-BASED-ENCRYPTION utilizado na implementação da Fase 2, utilizado na interação entre o proxy e o KDC (Ticket-Line)	
O anterior algoritmo pode ser parametrizável e forma testados outros ? Quais ?	

--	--

7.3 Ciphersuites configuráveis (ficheiro *ciphersuites.conf*) que foram testadas na implementação da FASE 2 (Não preencher se não foi implementado)

Configurações criptográficas estabelecidas dinamicamente através do KDC (*KDC Ticket-Line*) e que estão associadas aos *Tickets* distribuídos pelo KDC para o estabelecimento do canal seguro de *stream* entre o *proxy* e o *Stream Server*, para acesso, recepção e reprodução de um *stream* solicitado pelo *proxy*.

Ciphersuite (Identifique, por ex., AES/CTR/PKCS5Padding)	MACs Identifique as construções HMAC ou CMAC utilizadas para a Ciphersuite utilizada no protocolo da Fase 1 (ou alternativamente funções seguras de Hash que tenham também disso utilizadas)	Tamanho (em bits) da Chave Criptográfica	Tamanho (em bits) da chave do MAC (HMAC ou CLAC)

Configurações criptográficas (iniciais), estabelecidas estaticamente entre o *proxy* o KDC (*KDC Ticket Line*)

Ciphersuite / Password-Based Encryption Scheme (Identifique)	MAC Identifique a onstrução HMAC ou CMAC utilizada)	Tamanho da Password utilizada na autenticação entre o proxy e o KDC	Tamanho (em bits) da chave do MAC (HMAC ou CLAC)

7.4 Protocolo de Autenticação e Distribuição de Parâmetros de Associações Criptográficas entre o proxy e o KDC (KDC - Ticket Line)

Apresente uma especificação para o protocolo que implementou (utilizando uma notação apropriada para formalização de referência do protocolo desenhado). Utilize uma notação compreensível, apresentando o formato das mensagens trocadas entre o *proxy* e o KDC (*Ticket Selling Service*), seguindo como exemplo a notação de protocolos criptográficos estudados na disciplina.

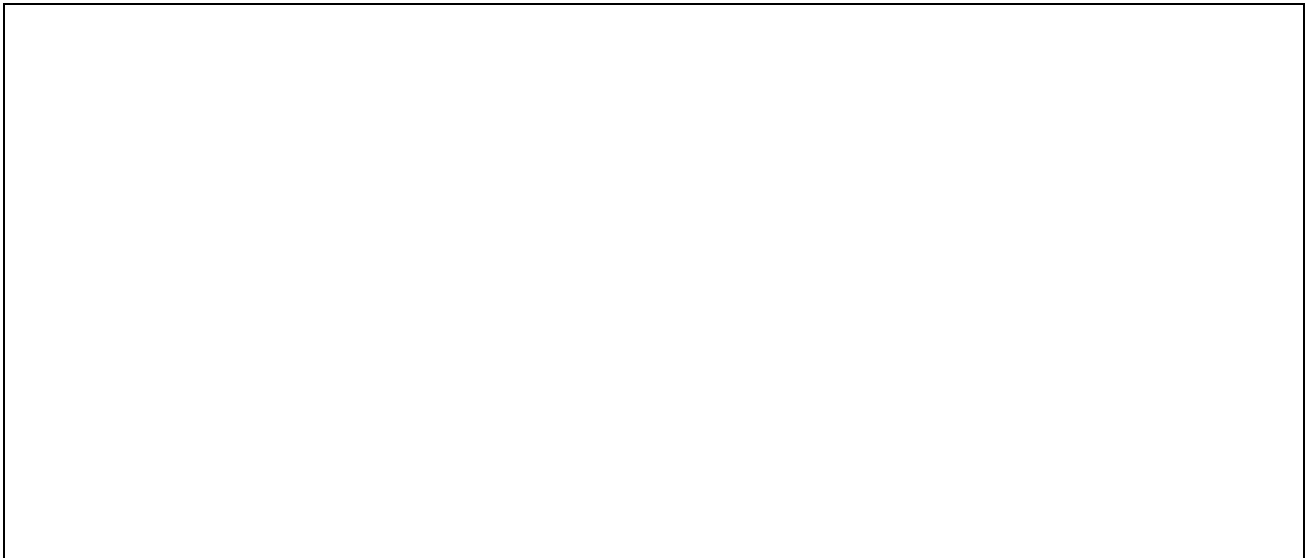
NOTA: Se incluiu esta especificação no Relatório do Trabalho, refira apenas: *Incluído no Relatório*.

	SIM	NAO
A implementação do anterior protocolo é suportada em UDP ?		
Sendo suportada em UDP, a implementação do protocolo faz uso (ou seja, suporta-se) como tipo de protocolo (<i>Protocol Type and Payload</i>) a partir do formato genérico de mensagens UDP subjacentes à utilização de <i>SecureDatagramSockets</i> , que são também usados (a partir da especificação da Fase 1) ? Isto é, o formato das mensagens seguras em canal UDP seguem ou estendem uma especificação do protocolo que também é usado para proteger as mensagens entre o <i>Proxy</i> e o <i>StreamServer</i> ?		
Se respondeu NAO à questão anterior, isso significa que o protocolo entre o <i>proxy</i> e o KDC (<i>KDC Ticket Line</i>) foi concebido e implementado com mensagens formatadas (de acordo com os componentes de segurança) de um modo completamente independente da especificação do protocolo da Fase 1 que protege a comunicação entre o <i>StreamServer</i> e o <i>Proxy</i> ?		

7.5 Concepção e implementação de Movie-Coins

Apresente a especificação (formato e proteção) que concebeu e que implementou para suportar MovieCoins e como é que as MovieCoins são validadas e geridas pelo KDC (Ticket Selling Service)

NOTA: Se incluiu esta especificação e respetiva informação no Relatório do Trabalho, refira apenas: *Incluído no Relatório.*



7.6 Concepção e implementação de Tickets

Apresente a especificação que concebeu e que implementou para suportar os Tickets que são distribuídos pelo KDC (Ticket Selling Service) na sequência da autenticação do proxy, e que este deve depois apresentar ao StreamServer para descarregamento e reprodução dos Streams.

NOTA: Se incluiu esta especificação e respetiva informação no Relatório do Trabalho, refira apenas: *Incluído no Relatório.*



7.7 Paralelismo da implementação na Fase 2

A sua implementação da Fase 2 permite que vários clientes em paralelo se autenticuem (via KDC ou *KDC-Ticket Line*) e uma vez autenticados e autorizados para receberem os parâmetros criptográficos possam estar reproduzir em paralelo diferentes *streams* (por exemplo em diferentes instâncias de VLC ou em computadores diferentes) ? Refira SIM ou NÃO e no caso de ser suportado explique como ou porquê.

7.8 Aspetos Valorativos da sua Implementação da Fase 2

Apresente ou argumente sobre quaisquer aspetos que considere valorativos para que sejam considerados na apreciação da sua implementação da Fase 2 do trabalho