DI/FCT/UNL
Mestrado Integrado em Engenharia Informática
Segurança de Redes e Sistemas de Computadores
1º Semestre, 2021/2022
Teste de Frequência nº 1 (27/Novembro/2021)

Código do TESTE: T1B-0927

Parte I – PARTE SEM CONSULTA

Questão 1

Considere que um principal (Alice) enviou a outro principal (Bob) a seguinte mensagem para proteger o *payload* M, em que estão assinaladas as partes correspondentes aos elementos para processamento criptográfico que foram concatenados na construção do envio dessa mensagem. A mensagem foi enviada por ALICE a BOB, suportada como *payload* de um pacote TCP

UID | | IV | | E_K (M | |SeqNr | | HMAC-SHA-256$_K$ (M)) | | HMAC-SHA1$_{Kmac}$ (D) | | X | | Y

$\underline{\quad A \quad}$ $\underline{\quad B \quad}$ $\underline{\quad C \quad}$ $\overline{X}$ $\overline{Y}$

$\underline{\quad D \quad}$

No envio da mensagem, foram usados os seguintes mecanismos criptográficos:

- UID: Identificador de Alice (*string*, passada em claro ou *plaintext*)
- IV : Vetor de inicialização de 128 bits (passado em claro ou *plaintext*
- E_K (P): Cifra simétrica AES do conteúdo P, usando o modo de operação GCM, com uma chave K de 256 bits e *padding* PKCS#5 (Nota: AES utiliza no processamento um bloco base de 128 bits)
- SeqNr : um número de sequência da mensagem (inteiro de 32 bits)
- Alice e Bob são supostos terem pares de chaves RSA com 1024 bits, conhecendo ambos as chaves públicas u do outro e confiando nas mesmas.
- HMAC-SHA256: utilizando a mesma chave usada para a cifra do componente F
- Kmac: é uma chave de 256 bits usada para o componente HMAC-SHA1 obtida por uma síntese da seguinte forma: SHA256 (H)
- V : Conjunto dos componentes *A e D* (concatenados)
- X: Corresponde ao resultado (bits) da assinatura digital do conjunto dos componentes A e F (concatenados), tendo sido usado uma assinatura RSA, tendo por base *padding* PKCS#1 e uma síntese SHA-512.
- Y : corresponde à cifra da chave K com a chave pública de Bob, ou seja, E_{KpubB} (K), sendo assim um "envelope" de distribuição da chave K por Alice a Bob. Para este caso foi usado RSA e *padding* OAEP.

Na análise dos componentes criptográficos indicados, considere a definição das propriedades de segurança com base na definição na *framework* X.800, responda às restantes alíneas: a) a g) **usando a folha para as suas respostas**

a) Preencha na tabela da folha de respostas (nas células da coluna identificada com "**INDICAÇÃO**") quais os componentes (A,B,C, D, X, Y ou NENHUMA) que associa a cada uma das propriedades de segurança indicadas na coluna "PROPRIEDADES"

b) Das seguintes subcategorização de propriedades de segurança como definidas na framework X.800, indique – assinalando na tabela da folha de respostas com **"V"**, as que são suportadas com base no formato criptográfico da especificação inicial no seu conjunto.

Perder-se-ia alguma ou algumas das propriedades de segurança? Qual ou quais?

c) Se em vez do formato inicial Alice enviar apenas a mensagem com as seguintes partes:

UID | | IV | | E_K (M | |SeqNr | | HMAC-SHA-256$_K$ (M)) | | Y

Perder-se-ia alguma ou algumas das propriedades de segurança? Qual ou quais?

d) Se em vez do formato inicial Alice enviar apenas a mensagem com as seguintes partes:

UID | | IV | | E_K (M | |SeqNr | | SHA-256 (M)) | | Y

Perder-se-ia alguma ou algumas das propriedades de segurança? Qual ou quais?

e) Se em vez do formato inicial Alice enviar apenas a mensagem com as seguintes partes:

UID | | IV | | E_K (M | |SeqNr) | | X | | Y

Perder-se-ia alguma ou algumas das propriedades de segurança? Qual ou quais?

f) Se em vez do formato inicial Alice enviar apenas a mensagem com as seguintes partes:

UID | | IV | | E_K (M | |SeqNr) | | X | | Y

g) Nas condições do formato inicial e independentemente do tamanho em bits de M, preencha a coluna RESPOSTAS" para cada uma das questões nas linhas da tabela na folha de respostas

Questão 2.

Alice e Bob possuem um par de chaves de curva elíptica. Alice conhece a chave pública de Bob, por ter obtido a mesma de forma confiável a partir de uma terceira parte (PKC ou CA). O mesmo se passa com Bob em relação à chave pública de Alice.

Alice e Bob querem estabelecer de forma segura uma chave de sessão Ks, com base num acordo que usa o método de *Diffie-Hellman*. Para tal vão usar parâmetros iniciais para um número primo ***P*** (de 2048 bits de representação), um valor inteiro ***G*** pré-definido com valor = 3 a usar como raiz primitiva do número ***P***. Desse modo estarão aptos a gerarem valores privados e públicos (***YpubA e YpubB***) para o acordo Diffie-Hellman, a partir de valores privados que cada um guardará de forma segura (***XprivA e XprivB***).

Se trocarem os valores públicos em claro, a chave de sessão que estabelecerão poderá ser comprometida por um adversário que atua como "homem-no-meio". Para o evitar:

a) Se a troca for feita de forma que apenas Alice enviar o seu número público ***YpubA*** assinado com uma assinatura ECDSA , mas Bob enviar ***YpubB*** a Alice em claro, isso já resolve o problema do potencial ataque do "homem no meio" ?

b) Deve a troca for feita de modo que cada um envie o respetivo valor público ao outro, cifrando com a chave pública EC do destinatário ? Justifique.

Questão 3.

Considere o sistema e protocolo Kerberos (V5) – abaixo representado. Responda às seguintes alíneas

(1) C → AS $Options \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) AS → C $Realm_c \| ID_C \| Ticket_{tgs} \| \mathrm{E}(K_c, [K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}])$

$Ticket_{tgs} = \mathrm{E}(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) C → TGS $Options \| ID_v \| Times \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C $Realm_c \| ID_C \| Ticket_v \| \mathrm{E}(K_{c,tgs}, [K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v])$

$Ticket_{tgs} = \mathrm{E}(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

$Ticket_v = \mathrm{E}(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| \mathrm{AD}_C \| Times])$

$Authenticator_c = \mathrm{E}(K_{c,tgs}, [ID_C \| Realm_c \| TS_1])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) C → V $Options \| Ticket_v \| Authenticator_c$

(6) V → C $\mathrm{E}_{Kc,v} [\, TS_2 \| Subkey \| Seq\# \,]$

$\mathrm{Ticket}_v = \mathrm{E}(K_v, [\mathrm{Flags} \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$

$Authenticator_c = \mathrm{E}(K_{c,v}, [ID_C \| Realm_c \| TS_2 \| Subkey \| Seq\#])$

(c) Client/Server Authentication Exchange to obtain service

a) Considera que o protocolo Kerberos garante segurança futura e passada perfeitas no contexto do estabelecimento de chaves criptográficas que são distribuídas através dos *Tickets Kerberos*? Porquê?

b) Que vantagem tem o uso combinado de NONCES no protocolo para além do mero uso de TIMESTAMPS e por que é que isso é importante do ponto de vista da segurança do protocolo?

c) Se o protocolo não for processado completamente com o envio da última mensagem (mensagem 6) enviada de um servidor V para um cliente C, já seria possível ter uma chave simétrica partilhada entre C e V. No entanto, que garantias de segurança se perderia pela não inclusão desta mensagem? Porquê?

d) Porque é que qualquer uma das variantes do protocolo Kerberos na especificação é vulnerável a ataques a *passwords* fracas, com base em ataques por dicionário?

DI/FCT/UNL
Mestrado Integrado em Engenharia Informática
Segurança de Redes e Sistemas de Computadores
1º Semestre, 2021/2022
Teste de Frequência nº 1 (27/Novembro/2021)

Parte II – PARTE COM CONSULTA

Questão 4

a) O processamento para cifrar, usando o algoritmo criptográfico simétrico *Tripe DES* pode usar uma chave de 168 bits e baseia-se nesse caso na sequência de 3 passos envolvendo o algoritmo DES em cada passo. A cifra é feita na sequência ENCRYPT-DECRYPT-ENCRYPT, (EDE) em que cada passo usa uma subchave de 56 bits. Diga se existe impacto na segurança se a sequência fosse ENCRYPT-ENCRYPT-ENCRYPT (EEE) e qual o interesse em termos a sequência EDE

b) Dada a sequência EDE para a cifra, indique qual a sequência que deve ser usada na decifra.

Questão 5

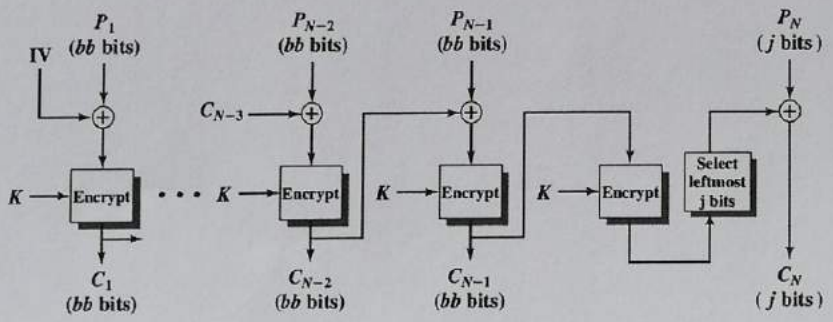
Suponha que vamos enviar um ficheiro usando transporte UDP e transmitindo o ficheiro por blocos de 1024 bits. Cada datagrama transporta como carga ou *payload* um número de sequência do bloco (Nseq, de tamanho 128 bits) e um bloco do ficheiro (*fileblock* de 1024 bits). O payload é enviado cifrado com o seguinte formato, usando-se uma cifra simétrica com parametrização de *padding* PKCS#5

Ciphertext Block $C_i = \{ Nseq_i \mid\mid fileblock_i \}_{KS}$

Usando-se a cifra em modo ECB se existir um erro na transmissão de um bloco i, apenas o correspondente bloco *plaintext* seria afetado na decifra. Mas por razões de segurança foi decidido usar CBC. Se houver um erro na transmissão de um dado C_i o bloco $fileblock_i$ e o bloco $fileblock_{i+1}$ não vão poder ser decifrados e assim obtidos corretamente no destino. Serão os restantes blocos $fileblock_{i+2}$ e posteriores também afetados? Justifique.

Questão 6

Considere o seguinte modo que pode ser usado para utilização com algoritmos de cifras de bloco.
Do ponto de vista de segurança este modo (que é conhecido coo modo CTS) garante um nível de segurança equivalente à do modo CBC. Que vantagem encontra na sua utilização ?



As seguintes Questões 7 e 8 estão relacionadas com análise de especificação do trabalho prático (PA#1)

Considere o contexto do trabalho prático (*Project Assignment #1*), os protocolos (SRTSP e SAPKDP) e os componentes da arquitetura da solução: *Media Player* usado pelos utilizadores, *ProxyBox* (PB) e real Time *Streaming Server* (RTSS) para disseminação de media (filmes) em canal protegido, para visualização em tempo real e Signaling Server

As respetivas especificações iniciais de referência dos protocolos SRTSP (especificação completa e simplificada – Fase 1) e SAPKDP podem ser encontradas em anexo, no caso de não as ter nos seus materiais de consulta.

Responda às seguintes questões, devendo notar que o pode fazer a partir do referencial da sua implementação especifica e respetivo potencial de configurações criptográficas suportadas, bem como (ou apenas) do seu entendimento que deve ser subjacente à análise das propriedades de segurança e mecanismos criptográficos usados na especificação de referência dos protocolos SRTSP (seja a versão completa da fase 2, seja da versão simplificada da Fase 1) e SAPKDP e independentemente da sua implementação ter endereçado a FASE 2 ou apenas a FASE 1.

Questão 7. Analisando o protocolo SRTSP (ANEXO) à luz das definições das propriedades de segurança estabelecidas na Framework conceptual X.800:

a) Diga se o protocolo SRTSP, **versões simplificada e completa**, asseguram proteção associada às seguintes propriedades de segurança. Pode justificar as suas respostas nos casos que entenda ser relevante.
   - A1) Confidencialidade do tráfego (*traffic flow confidentilaity*)
   - A2) *Connectionless confidentiality*
   - A3) *Connection-confidentiality*
   - A4) Connectionless Integrity w/ Recovery
   - A5) Connectionless Integrity without Recovery
   - A6) Connection Integrity
   - A7) Peer-authentication
   - A8) Non-Repudiation of frames sent by the StreamServer
   - A9) Access Control
   - A10) Non-Replaying

b) No protocolo SRTSP haveria vantagem em que todas as provas de integridade (*IntCheck*) nas diversas mensagens do protocolo seja feito apenas com uma função de síntese (*Secure Hashing*) em vez de usar MACs (ex: funções HMAC) ? Argumente.

c) Faça uma proposta de uma possível otimização do protocolo para a versão completa (mantendo as mesmas propriedades de segurança) se a configuração criptográfica envolve sempre a utilização do modo GCM no caso da proteção das mensagens 2,3,4,5,6 , indicando o que poderia ser otimizado.

Questão 8

Partindo da análise do protocolo SAPKDP (ANEXO) apresente uma variante do protocolo com uma proposta de especificação de modo que as chaves de sessão estabelecidas dinamicamente para as configurações criptográficas simétricas (e subsequente distribuição segura das mesmas ao componente *StreamServer* no contexto do protocolo SRTSP) seja gerada com base num acordo contributivo usando o método DIFFIE-HELLMAN, de modo a garantir os seguintes requisitos:

- R1: Apenas as mensagens (*Msg types*) 4, 5 e 6 do protocolo SAPKDP podem ser alteradas
- R2: A chave de sessão (*Seccion Key*) na mensagem 6, passa a ser calculada com base num acordo contributivo, com envolvência do PROXY e do SIGNALING SERVER
- R3: A solução deve ser segura contra um potencial adversário atuando como homem no meio ou que tente atacar qualquer uma das mensagens do protocolo, mantendo-se assim as propriedades de segurança inciais.
- R4: Cada nova execução do protocolo SAPKDP deve garantir a a geração e estabelecimento de uma nova chave de sessão para a posterior visualização dos filmes (*media contents*), com garantias de segurança futura e passada perfeitas

Para a sua resposta devera apresentar as suas novas especificações propostas, representando as mensagens alteradas (4,5 e 6), a sua nova formatação proposta e os respetivos elementos criptográficos. Utilize uma notação similar **que clarifique bem o que propõe** seguindo uma notação de referência inspirada na especificação inicial) que deve complementar com uma legenda que descreva sumariamente os elementos criptográficos na notação utilizada.

SRTSP: Secure Real Time Streaming Protocol

- Protocolo que envolve os componentes *ProxyBox* (PB) e *Real Time Streaming Server* (RTSS) e respetivos *endpoints* de comunicação.
- Implementado como camada de segurança suportada no transporte UDP e assim permitindo a disseminação de media frames, em tempo real, com possível adopção de IP *Unicasting* ou IP *Multicasting*.

Ent.Flow	Message description	M. Type	Specifiction
PB > RTSS Round 1	PB-RequestAndCredentials	1	{ IP, Port, ciphersuiteConf, CryptoSA, SessionKey, MacKey, NC1 }$_{K_{pubRTSS}}$, Na1 , ECDSASignature$_{K_{privSS}}$ (Payloads), Intcheck1
RTSS > PB Round 2	RTSS-Verification	2	{ Na1', Na2, TickeyValidityConfirmation}$_{K_S}$, Intcheck2
PB > RTSS Round 3	PB-AckVertification	3	{ Na2, Na3}$_{K_S}$, Intcheck3
RTSS > PB Round 4	RTSS-SynkInitialFrame	4	{ Na3, initmark-frame, _}$_{K_S}$, IntCheck4
RTSS > PB Rounds i	EncryptedStreamData	i	Encrypted Stream Data (Media Frames) {SequenceNumber, Frame }$_{K_S}$, InitCheckF
Round N	RTSS-SynkFinalFrame	N	{endmark-fram}$_{K_S}$, IntCheckN

Note que na especificação acima prevê-se a utilização de assinaturas de curva elíptica (ECDSA) na mensagem 1 (*Msg Type* 1) e o processamento da mensagem 1 (*Msg type* = 1) decorre da execução completa anterior do protocolo SAPKDP entre o Proxy (PB) e o *Signaling Server* e que está representado abaixo.

SRTSP: *Secure Real Time Streaming Protocol*

(Versão Simplificada – FASE 1 de Implementação)

Note que a versão simplificada do protocolo, (sem implementação completa do protocolo SRTSP, se apenas foi esta a sua implementação) é baseada no envio pelo *StreamServer* para o *ProxyBox* de *media frames* protegidas conforme a especificação da mensagem i (rounds i) e com uso de configurações estáticas nos *endpoints* dos principais StreamServer e *ProyyBox*.

SAPKDP: Secure Authentication, Payment and Key Distribution Protocol

Ent.Flow	Message description	M. Type	Functional Description
PB > SS Round 1	PB-Hello	1	UseID, ProxyBoxId
SS > PB Round 2	SS-AuthenticationRequest	2	N1, Salt, Counter
PB > SS Round 3	PB-Authentication	3	PBE$_{User\&Pwd,Salt,Counter}$ (N1',N2,MovidID), IntCheck3
SS > PB Round 4	SS-PaymentRequest	4	ECDSASignature$_{K_{privSS}}$ (Price, N2', N3), IntCheck4 (1)
PB > SS Round 5	PB-Payment	5	ECDSASignature$_{K_{privPB}}$ (N3', N4, PaymentCoin), IntCheck5 (2)
SS > PB Round 6	SS-TicketCredentials	6	{IP, Port, MovieID, ciphersuiteConf, CryptoSA, SessionKey, MacKey, N4' }$_{K_{pubPB}}$, {IP, Port, MovieID, ciphersuiteConf, CryptoSA, SessionKey, MacKey, NC1 }$_{K_{pubRTSS}}$, ECDSASignature$_{K_{privSS}}$ (Payloads), IntCheck6