Segurança e Aplicações de Hardware Confiável 21/22 – Época Normal – 27/6/2022

Duração: 2h

Grupo 1 - Smart Cards

1.1. Definições.

a) Uma das distinções chave entre os smart cards modernos e os cartões de fita magnética é a capacidade de armazenamento segura. Dê um exemplo de ataque que acontece trivialmente em cartões de fita magnética mas não em smart cards modernos, devido à ausência desta garantia.

b) Dê dois exemplos de caraterísticas essenciais associadas a smart cards, para além da capacidade para armazenamento seguro.

1.2. Autenticação.

A realização de operações seguras em smart cards é habitualmente precedida por uma sequência de validações do cartão perante o leitor, e vice-versa.

a) Explique a importância de autenticar o leitor de cartões (i.e. demonstrar que o leitor é genuíno), no contexto de realizar operações em smart cards.

b) Dê um exemplo de uma aplicação vulnerável a ataques, caso o cartão não seja validado perante um leitor de cartões (i.e. demonstrar que o cartão é genuíno).

1.3. Comunicação e APDUs.

a) Explique o que significa descrever a comunicação entre smart cards e leitores como sendo *half-duplex*.

b) Após inicialização e seleção do applet, suponha que envio o seguinte APDU para o cartão: 0x14, 0x0A, 0x01, 0x02, 0x01, 0x00. O seguinte APDU é recebido como resposta: 0x90, 0x00. Enumere as diferentes componentes desta comunicação – o que representa cada um destes bytes?

c) Esta troca de APDUs está correta? Justifique.

1.4. Memória e Armazenamento Confiável.

a) Estudamos quatro tipos diferentes de memória presentes em smart cards: RAM, ROM, EEPROM e Flash. Associe as seguintes caraterísticas a estes diferentes tipos de memória (responda na folha de exame).

Nota: Algumas caraterísticas são comuns a várias memórias. Nesse caso, indique apenas uma.

1. Persistente e mutável
2. Tempo de vida (nº de acessos) limitado
3. Utilizada para armazenamento de programas fixos
4. Informação não é preservada sem energia
5. Pode ser acedida um número ilimitado de vezes

Grupo 2 - HSMs e TPMs

2.1. Inicialização Confiável.

a) Os protocolos estudados para inicialização confiável na TPM pressupõem a existência de um valor sucinto (uma hash) para validar integridade do código de inicialização da plataforma. Porque é que o conseguimos fazer apenas através de uma hash, em alternativa a ter o código todo dentro da TPM?

b) Um componente essencial do TPM é o seu gerador de números/valores aleatórios. Que ataques podem surgir se um TPM tiver que obter a sua aleatoriedade através do sistema operativo?

2.2. Hardware Security Modules.

a) Dê dois exemplos de benefícios de utilizar APIs como a Cryptoki para aceder a HSMs, ao invés de permitirmos que cada tecnologia de HSM defina a sua própria API.

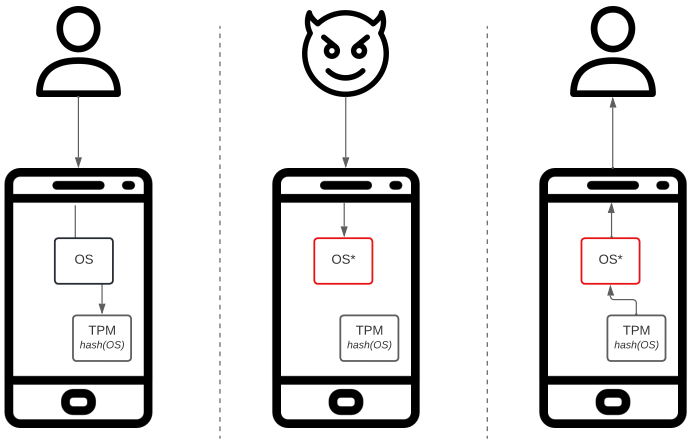
b) Indique se as seguinte frase está correta, justificando.

Tendo em conta que HSMs dão garantias elevadas de segurança para operações criptográficas, estas devem também ser consideradas para outro tipo de aplicações. Por exemplo, eu posso usar HSMs para armazenar e fazer cálculos sobre a informação bancária da minha empresa.

2.3. Assunções

Considere um sistema desenhado para proteger a integridade de um sistema operativo para um dispositivo móvel. A sua utilização divide-se em 3 etapas (na figura, da esquerda para a direita):

- Numa primeira etapa, existe um setup confiável, onde o utilizador inicializa um sistema operativo confiável (OS), e armazena o seu measurement no TPM ($hash(OS)$).
- Numa segunda etapa, o sistema fica vulnerável a ataques da parte do adversário. Este pode corromper o sistema operativo, alterando o seu código (OS^*). Porém, este não consegue alterar os valores internos da TPM.
- Finalmente, o utilizador volta a ter controlo do seu sistema. Pode chamar a TPM para obter o measurement, de forma a verificar a integridade do sistema operativo.



a) Que garantias é que o TPM está a dar ao utilizador, no que toca à integridade do seu sistema operativo?.

b) Considere, agora, que o adversário também consegue corromper a memória do TPM, podendo alterar os valores armazenados. Dê um exemplo de ataque que subverte o sistema para a instalação de um sistema operativo corrupto.

c) Se quisermos adaptar este sistema para uma arquitetura x86, conseguimos ter as mesmas garantias se a nossa solução de hardware seguro for Intel SGX? Justifique.

Grupo 3 - Intel SGX e ARM TrustZone

3.1. Ambientes de Execução Isolados

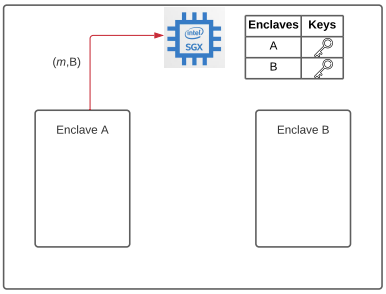
a) Migrar aplicações para serem executadas remotamente é uma abordagem apelativa do ponto de vista de preço, performance e escalabilidade. Desta forma, não temos que nos preocupar com adquirir e gerir a infraestrutura física, e podemos pagar apenas a computação que as nossas aplicações precisam. Indique **um problema de segurança** associado a este modelo, onde os dados são preservados remotamente, e descreva **como este pode ser resolvido** utilizando hardware confiável.

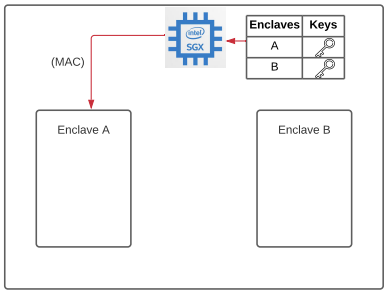
b) Indique se as seguintes frases são verdadeiras ou falsas (responda na folha de exame).

1. Intel SGX e ARM TrustZone são importantes, pois não existe forma de fazer computação sobre dados protegidos usando apenas software.
2. É possível programar as funcionalidades criptográficas de um HSM num enclave de Intel SGX.
3. Código a correr no mundo seguro do ARM TrustZone é resistente a ataques de timing.
4. Executar programas em ambientes isolados é uma medida que me protege contra potenciais bugs dos mesmos programas.
5. É possível converter qualquer programa desenhado para ARM TrustZone para ser executado usando uma TPM.

3.2. Intel SGX

Considere a seguinte imagem, que descreve um passo do mecanismo de atestação intra-plataforma do Intel SGX. O enclave A deseja produzir uma atestação de uma mensagem m para o enclave B. Como resposta a este pedido, o CPU devolve um MAC da mensagem m e do código do enclave A, criado utilizando a chave associada ao enclave B.





a) Podemos redesenhar o Intel SGX para devolver a chave do enclave B ao enclave A, de modo a que possa ser o próprio enclave a produzir o MAC em questão? Justifique.

b) Porque é que não podemos simplesmente utilizar o mecanismo existente de atestação intra-plataforma para atestar valores para um cliente que esteja a comunicar externamente com a plataforma (inter-plataforma)?

3.3. ARM TrustZone.

a) No mundo seguro do ARM TrustZone, só pode ser executado código assinado por uma entidade de confiança. Indique uma desvantagem associada a esta restrição.

b) Qual é a vantagem de equipar sistemas de ARM TrustZone com dois sistemas operativos: um sistema operativo na zona de memória confiável (Trusted OS), e um sistema operativo na memória normal (Untrusted OS)?

Grupo 4 - Ataques e Contramedidas

4.1. Modelo Embebido

a) Porque é que o adversário contra hardware confiável tende a ser consideravelmente mais poderoso que um adversário contra um sistema clássico?

b) Explique o que são ataques físicos, e de que forma se comportam de forma diferente da criptanálise.

4.2. Vulnerabilidades e Contramedidas

Considere uma função para deteção de vulnerabilidades, que se comporta da seguinte forma:

- Recebe uma lista de especificações de sistema (em formato de array)
- Num loop, itera sobre as várias entradas de uma base de dados de vulnerabilidades (BD):
  - Se detetar que uma (ou mais) especificações recebidas estão associados a uma vulnerabilidade na BD, termina e retorna uma mensagem de "vulnerabilidade detetada".
  - Caso contrário, segue para a próxima entrada na base de dados.
- Se o ciclo terminar, nenhuma vulnerabilidade foi associada, e o programa termina com uma mensagem de "sucesso".

a) Assuma agora que o tamanho da base de dados de vulnerabilidades é público. Explique como é possível, através de um *timing attack*, distinguir (com algum nível de confiança) se o resultado desta operação foi "vulnerabilidade detetada", ou "sucesso".

b) Explique como ataques de manipulação de energia podem ser usados para atacar um sistema de hardware confiável.

c) Desativar a execução especulativa é uma forma natural e eficaz de evitar ataques através deste mecanismo. Porque é que esta medida não é tomada em todos os sistemas para os proteger contra esta fragilidade?

d) Uma disciplina conhecida para prevenir ataques de timing é a de constant-time. Descreva que garantias temos de um programa que corre em tempo constante.

e) Descreva os benefícios principais de exigir certificação de hardware, quando queremos desenvolver uma aplicação que visa utilizar hardware confiável como âncora de confiança.