Instituto Superior de Engenharia de Lisboa
Mestrado de Engenharia Informática e Multimédia

Ano letivo 2025/2026

Cibersegurança

Trabalho Prático nº 3

Turma

Miguel Azenha, A47708
Marta,

Dezembro de 2025

Índice

1 Introdução **1**

2 Enquadramento Teórico **2**
2.1 Trusted Execution Environments 2
2.2 Intel Software Guard Extensions 2
2.3 Arquitetura dos Enclaves 2

3 Interface entre Código Confiável e Não Confiável **3**

4 Selagem e Armazenamento Seguro **3**

5 Minimização da Trusted Computing Base **3**

6 Síntese **3**

7 Implementação **5**

8 Arquitetura Geral da Aplicação **5**

9 Particionamento entre Domínios Confiável e Não Confiável **5**

10 Estruturas de Dados e Tipos Implementados **6**

11 Interface EDL e Comunicação entre Domínios **6**

12 Operações Sensíveis no Enclave **7**
12.1 Verificação e Política de Passwords 7
12.2 Geração de Passwords Seguras 7
12.3 Manipulação da Carteira 7

13 Selagem e Desselagem de Dados **7**

14 Minimização da Trusted Computing Base **8**

15 Fluxo Completo das Operações **8**

16 Resumo **8**

17 Conclusões **10**

Índice de figuras

1 Introdução

O presente trabalho prático tem como objetivo explorar a tecnologia Intel Software Guard Extensions (SGX) no desenvolvimento de uma aplicação de carteira eletrónica, destinada ao armazenamento seguro de credenciais de acesso a plataformas informáticas. A segurança e confidencialidade das credenciais são fatores críticos, dado que estas contêm informações sensíveis como nomes de utilizador, senhas de acesso e descrições dos contextos de utilização.

A aplicação a desenvolver deverá permitir a gestão de até 100 credenciais diferentes, disponibilizando funcionalidades essenciais ao utilizador, tais como:

1. criação de uma nova carteira;
2. visualização do conteúdo da carteira;
3. adição de novas credenciais;
4. remoção de credenciais existentes;
5. geração de senhas de acesso seguras, com comprimento variável entre 8 e 100 caracteres.

Para cada credencial, devem ser armazenados o nome de utilizador (até 100 caracteres), a senha de acesso (8 a 100 caracteres) e uma descrição identificativa do cenário de utilização (até 100 caracteres). A persistência dos dados deverá ser garantida através de armazenamento em ficheiro, utilizando cifra AES-GCM de 128 bits para assegurar confidencialidade e integridade, incluindo a senha de acesso que protege a carteira.

Adicionalmente, a funcionalidade de *sealing* deverá ser utilizada para garantir a proteção dos dados no ficheiro, permitindo a sua recuperação segura em execuções posteriores. A minimização da *Trusted Computing Base* (TCB) é recomendada, de forma a reduzir a superfície de ataque e aumentar a segurança global da aplicação.

Este trabalho permite aplicar na prática os conceitos aprendidos em aula, incluindo:

- criação e destruição de enclaves SGX;
- comunicação segura entre aplicação não confiável e enclave (*ECALLs* e *OCALLs*);
- implementação de mecanismos de confidencialidade e integridade em dados persistentes;
- utilização de técnicas de selagem (*sealing*) para armazenamento seguro de dados sensíveis.

A realização deste projeto reforça a compreensão da utilização de enclaves SGX em cenários de segurança real e a importância da proteção de dados sensíveis em aplicações práticas.

2 Enquadramento Teórico

Este capítulo apresenta os fundamentos teóricos necessários para compreender o desenvolvimento de aplicações baseadas em *Trusted Execution Environments* (TEEs), com particular foco na tecnologia Intel Software Guard Extensions (SGX), utilizada neste trabalho para proteger dados sensíveis de uma carteira eletrónica. São explorados os conceitos fundamentais de enclaves, modelo de ameaça, comunicação segura entre domínios de confiança e mecanismos de selagem (*sealing*) para armazenamento persistente.

2.1 Trusted Execution Environments

Um *Trusted Execution Environment* (TEE) é um ambiente seguro, isolado do sistema operativo e das aplicações comuns, que se destina à execução de código sensível. O objetivo dos TEEs é proteger dados em uso, enfrentando uma limitação de segurança tradicional, embora a criptografia proteja dados em repouso e em trânsito, estes permanecem vulneráveis quando estão a ser processados na memória de sistemas potencialmente comprometidos.

Os TEEs garantem:

- **Confidencialidade**: dados dentro do TEE não podem ser acedidos por software externo
- **Integridade**: o código e dados protegidos não podem ser modificados
- **Isolamento**: o estado do TEE é independente do sistema operativo
- **Autenticidade**: o TEE permite provar a sua identidade e integridade

SGX e TrustZone representam abordagens distintas dentro do mesmo paradigma de computação confiável, o SGX fornece enclaves ao nível de aplicação, enquanto TrustZone divide todo o processador em duas áreas, uma segura e outra não segura.

2.2 Intel Software Guard Extensions

Intel SGX é uma tecnologia de isolamento de memória que permite criar enclaves, regiões protegidas que executam código de forma isolada do sistema operativo, BIOS, firmware, hipervisor ou aplicações externas. O mecanismo baseia-se na Enclave Page Cache (EPC), uma área de memória física protegida por hardware e inacessível a processos externos, mesmo com privilégios elevados.

2.3 Arquitetura dos Enclaves

Um enclave é uma biblioteca compilada como *shared object* contendo:

- **Código confiável**;
- **Dados sensíveis**;
- **Metadados de segurança**;
- **Tamanho do heap**, **stack** e número de **trusted threads**.

Estes metadados são utilizados pelo carregador não confiável para preparar a EPC, mas sem nunca aceder ao conteúdo real do enclave.

De acordo com a Intel, assumem-se como não confiáveis o sistema operativo, hipervisor, drivers,

firware e aplicações externas. O atacante pode controlar todo o sistema operativo e observar a memória, mas não consegue let ou alterar os dados dentro do enclave, pois eles estão protegidos por hardware.

3 Interface entre Código Confiável e Não Confiável

A comunicação entre a aplicação não confiável e o enclave é feita com base em:

ECALLs chamadas da aplicação para o enclave (entrada);

OCALLs chamadas do enclave para a aplicação (saída).

Esta interface é descrita no ficheiro `.edl`, que especifica:

- tipos de dados permitidos atravessar a fronteira de confiança;
- tamanhos de buffers;
- direções de ponteiros;
- validação automática feita pelo proxy gerado pelo `sgx_edger8r`.

4 Selagem e Armazenamento Seguro

A selagem (*sealing*) é um mecanismo essencial para permitir persistência de dados confidenciais. Embora o enclave ofereça proteção em memória, os seus dados são perdidos quando o mesmo é encerrado. Assim, é necessário guardar estes dados num ficheiro ou base de dados externa, mas de forma criptograficamente protegida antes de os dados deixarem o enclave.

5 Minimização da Trusted Computing Base

Um dos princípios fundamentais na construção de TEEs é a minimização da TCB — o conjunto de código confiável cuja segurança é crítica. Tal como discutido nas aulas, um enclave deve conter:

- apenas a lógica que manipula dados sensíveis;
- apenas funções essenciais;
- estruturas de dados estritamente necessárias.

No projeto:

- toda a criptografia, verificação de passwords, manipulação da carteira e geração segura de passwords ocorrem no enclave;
- a interface é reduzida a um conjunto mínimo de ECALLs.

6 Síntese

O enquadramento teórico apresentado demonstra que Intel SGX fornece um TEE adequado para proteção de dados sensíveis ao nível da aplicação, oferecendo isolamento forte e suporte nativo para

selagem. Estas capacidades tornam SGX ideal para a construção de uma carteira segura persistente, tal como implementado neste trabalho.

7 Implementação

Este capítulo descreve detalhadamente a implementação da aplicação de carteira eletrónica desenvolvida com suporte da tecnologia Intel SGX. São analisadas as decisões de particionamento do código entre os domínios confiável (enclave) e não confiável (aplicação), o desenho da interface de comunicação entre estes domínios, a utilização das operações de selagem para armazenamento persistente e as medidas adotadas para garantir a minimização da *Trusted Computing Base* (TCB).

8 Arquitetura Geral da Aplicação

A aplicação desenvolvida segue o modelo padrão para aplicações Intel SGX, composto por duas componentes principais:

- **Aplicação não confiável** (*untrusted*, ficheiro `app.c`): responsável pela interação com o utilizador, gestão de argumentos, criação e destruição do enclave e execução de OCALLs relacionadas com operações externas (e.g. I/O em ficheiros e consola).
- **Enclave SGX** (`enclave.c`): contém todas as operações sensíveis, nomeadamente gestão da carteira, verificação de passwords, geração de senhas aleatórias, validação de políticas de segurança, além das operações de selagem e desselagem.

Este modelo segue a recomendação presente nas aulas: *colocar no enclave apenas o mínimo código necessário para manipular dados sensíveis*, mantendo toda a lógica genérica na aplicação não confiável, de forma a reduzir a superfície de ataque.

9 Particionamento entre Domínios Confiável e Não Confiável

Com base nos princípios apresentados nos slides de *Writing an SGX Application*, foi realizada uma análise sistemática dos dados e funções que exigiam proteção forte. Foram classificados como **sensíveis**:

- a **master password** da carteira;
- todas as **credenciais** (títulos, utilizadores e passwords associadas);
- operações de **verificação** de passwords;
- o algoritmo de **geração de senhas** aleatórias;
- a **estrutura interna** da carteira (`wallet_t`).

Estes elementos foram colocados no enclave, que garante:

- isolamento em memória (EPC);
- impossibilidade de leitura pelo SO ou hipervisor;
- integridade do código carregado;
- execução apenas de código analisado.

Por outro lado, operações consideradas não sensíveis foram mantidas na aplicação externa:

- parsing de argumentos da linha de comandos;
- impressão de mensagens na consola;
- gestão de ficheiros com dados selados;
- validação sintática de parâmetros de entrada.

Esta separação corresponde diretamente ao modelo apresentado nos slides, onde o código com acesso a dados críticos deve residir numa TCB reduzida.

10 Estruturas de Dados e Tipos Implementados

A carteira é armazenada no enclave como uma instância da seguinte estrutura:

```
typedef struct {
    char title[WALLET_MAX_ITEM_SIZE];
    char username[WALLET_MAX_ITEM_SIZE];
    char password[WALLET_MAX_ITEM_SIZE];
} item_t;

typedef struct {
    char master_password[WALLET_MAX_ITEM_SIZE];
    size_t size;
    item_t items[WALLET_MAX_ITEMS];
} wallet_t;
```

O enclave contém a única cópia destas estruturas em memória clara, garantindo que:

- nenhuma credencial é carregada na aplicação externa;
- a master password nunca sai da EPC;
- apenas dados selados (ciphertext + metadados GCM) atravessam a fronteira do enclave.

11 Interface EDL e Comunicação entre Domínios

Seguindo as regras do *Enclave Definition Language* (EDL), a interface exposta pelo enclave foi reduzida ao conjunto mínimo necessário de ECALLs:

- `ecall_create_wallet()`
- `ecall_show_wallet()`
- `ecall_add_item()`
- `ecall_remove_item()`
- `ecall_change_master_password()`
- `ecall_generate_password()`

Conforme recomendado nas aulas, limitar o número de ECALLs reduz o ataque potencial ao enclave,

uma vez que cada fronteira cruzada representa um ponto de validação.

As OCALLs implementadas também seguem o princípio de minimização:

- `ocall_print_string()` — impressão segura na consola;
- `ocall_save_sealed_data()` — gravação de dados selados;
- `ocall_load_sealed_data()` — leitura de dados selados.

A EDL gerada por `sgx_edger8r` trata automaticamente:

- validação de buffers;
- marshalling seguro;
- prevenção de *buffer overflows*;
- verificação de direções de ponteiros.

12 Operações Sensíveis no Enclave

12.1 Verificação e Política de Passwords

A função `check_password_policy()` implementa restrições minimamente necessárias para evitar entradas inseguras. Colocá-la dentro do enclave impede ataques baseados na manipulação do fluxo de validação externo.

12.2 Geração de Passwords Seguras

A função `ecall_generate_password()` utiliza diretamente `sgx_read_rand()`, que acede ao gerador de números aleatórios seguro do processador. O uso de TRNG é obrigatório para evitar previsibilidade e ataques criptográficos, substituindo funções pseudo-aleatórias tradicionais.

12.3 Manipulação da Carteira

Todas as funções que alteram o conteúdo da carteira operam exclusivamente dentro do enclave:

- cópia e verificação de campos;
- verificação da master password;
- adição e remoção de itens;
- substituição da master password.

13 Selagem e Desselagem de Dados

A persistência dos dados é garantida por selagem utilizando as funções do SDK:

- `sgx_seal_data()`;
- `sgx_unseal_data()`.

Os dados selados são armazenados num ficheiro externo, mas:

- incluem GCM authentication tag;
- incorporam IV gerado automaticamente;
- são encriptados com chave derivada pelo *hardware*;
- são autenticados criptograficamente.

Estes aspetos correspondem diretamente ao slide "Sealing Properties".

14 Minimização da Trusted Computing Base

Durante o desenvolvimento, foram tomadas medidas específicas para reduzir ao máximo a TCB:

- nenhuma operação de I/O dentro do enclave;
- nenhuma alocação dinâmica complexa fora das funções permitidas;
- ausência de bibliotecas externas não confiáveis dentro do enclave;
- validação estrita de parâmetros em cada ECALL;
- redução da interface exposta no EDL ao mínimo.

Estas decisões seguem as recomendações do slide: *"ISVs should attempt to minimize the enclave size."*

15 Fluxo Completo das Operações

As principais operações seguem o fluxo:

1. Aplicação recebe input do utilizador.
2. ECALL é invocada com dados sanitizados.
3. Enclave valida parâmetros e executa operação.
4. Enclave sela dados (apenas quando necessário).
5. OCALL guarda dados selados no ficheiro.
6. Enclave devolve resultado seguro.

Em nenhum momento dados sensíveis são expostos ao exterior.

16 Resumo

A implementação cumpre todos os objetivos do trabalho:

- protege credenciais e passwords dentro de um enclave SGX;
- assegura confidencialidade e integridade via AES-GCM;

- usa selagem para armazenamento persistente;
- reduz a superfície de ataque via TCB minimizada;
- implementa comunicação segura via ECALLs/OCALLs;
- utiliza TRNG para criação de passwords.

Este capítulo demonstra que a solução desenvolvida segue as melhores práticas ensinadas nas aulas e documentadas pela Intel.

17 Conclusões

O trabalho desenvolvido permitiu aplicar, de forma prática e aprofundada, os conceitos estudados na unidade curricular no âmbito da utilização de *Trusted Execution Environments* (TEEs), com particular foco na tecnologia Intel SGX. A implementação de uma carteira eletrónica capaz de armazenar credenciais sensíveis demonstrou concretamente como os enclaves podem reforçar a proteção de dados críticos em sistemas potencialmente comprometidos.

A arquitetura final da aplicação seguiu rigorosamente as recomendações apresentadas tanto nas aulas como na documentação da Intel. Em particular, foi dada especial atenção à **minimização da Trusted Computing Base**, mantendo dentro do enclave apenas o conjunto estritamente necessário de funções e dados: a master password, as credenciais, as operações de validação, manipulação e geração de dados sensíveis. Todas as operações de interação com o exterior—tanto com o utilizador, como com o sistema de ficheiros—foram mantidas na aplicação não confiável, reduzindo a superfície de ataque e aumentando a resiliência do sistema.

Um dos aspetos mais relevantes do projeto foi a aplicação prática das técnicas de **selagem** e **desselagem**, essenciais para garantir persistência dos dados num formato seguro. A utilização das primitivas `sgx_seal_data()` e `sgx_unseal_data()` permitiu assegurar que a carteira nunca é armazenada em memória ou em disco de forma legível, sendo protegida por cifragem AES-GCM a 128 bits com chaves geradas e geridas internamente pelo hardware SGX. Esta propriedade é fundamental para proteger o sistema mesmo em cenários de comprometimento total do sistema operativo ou de acesso físico ao dispositivo.

O trabalho possibilitou também a experimentação com os mecanismos de comunicação entre o enclave e a aplicação, nomeadamente através de ECALLs e OCALLs. A correta definição da interface EDL revelou-se essencial para assegurar o encapsulamento dos dados sensíveis e a validação rigorosa de parâmetros nas transições entre domínios. Este modelo reforça a compreensão das limitações e desafios específicos dos TEEs, especialmente no que diz respeito ao controlo apertado de fronteiras e à necessidade de evitar a exposição acidental de dados.

Outro contributo importante foi a integração do gerador de números aleatórios seguro do SGX para a geração de passwords fortes. Esta componente demonstrou na prática a importância de se recorrer a fontes de entropia confiáveis, uma vez que mecanismos pseudo-aleatórios tradicionais não seriam adequados para um cenário de segurança elevada como o presente.

Apesar de bem-sucedida, a implementação apresentou algumas limitações intrínsecas à tecnologia SGX, tais como o tamanho reduzido da memória EPC e a impossibilidade de utilizar certas bibliotecas ou primitivas avançadas no interior do enclave. Estas limitações foram contornadas através de um desenho modular e cuidadoso, mas refletem ainda assim a realidade de que os enclaves exigem um planeamento rigoroso da arquitetura e dos fluxos de dados.

Em suma, o projeto permitiu consolidar uma compreensão sólida sobre:

- a importância de isolar dados e operações sensíveis;
- o papel dos enclaves como mecanismos de proteção mesmo perante um sistema operativo comprometido;
- a utilização prática das operações de selagem para garantir persistência segura;
- as dificuldades e decisões envolvidas na minimização da TCB;

- o modelo de comunicação seguro baseado em ECALLs e OCALLs;
- o impacto das restrições dos TEEs no desenho de software seguro.

A solução final cumpre todos os requisitos especificados, apresentando uma carteira eletrónica funcional, segura e alinhada com as melhores práticas recomendadas para o uso da tecnologia Intel SGX. O trabalho realizado constitui uma base sólida para o desenvolvimento de sistemas mais complexos de gestão e proteção de dados sensíveis, seja no contexto de aplicações de segurança, criptografia ou computação confidencial.