Tamper-Proof Digital Vault

Fábio Santos 118351 Rodrigo Marques 118587

10 de maio de 2024

Conteúdo

1 Introdução

Este projeto aborda a implementação de um Trusted Personal Data Vault (TPDV) utilizando o Intel SGX (Software Guard Extensions). O TPDV permite aos utilizadores armazenar e gerir Assets de forma segura, garantindo confidencialidade e integridade dos dados através do uso de enclaves seguros.

No contexto do Intel SGX, os enclaves são áreas protegidas na memória onde operações sensíveis podem ser executadas de forma isolada, protegendo os dados contra acessos não autorizados, mesmo por parte do sistema operativo. Neste trabalho, exploramos várias funcionalidades do TPDV, desde a criação e adição de Assets até operações avançadas como clonagem e verificação de integridade dos dados.

2 Funcionalidades

2.1 Criar um TPDV

Um TPDV é um ficheiro binário que irá guardar Assets introduzidos por um utilizador. A sua estrutura é constituída por um Header, mostrado na Figura 1, e os respetivos Assets. O Header é constituído pelo nome do autor, a palavra passe do TPDV, o número total de Assets que ele contém e um nonce (number used only once). O nonce é um número usado para verificar se a integridade do TPDV foi comprometida, uma vez que ele representa os últimos 4 bytes do Hash do conjunto dos Assets do TPDV. Existe uma desvantagem nesta abordagem: uma vez que o Header não é tido em consideração, a integridade do nome do autor, palavra passe e número de Assets não é verificada (o Nonce não pode participar no digest, pois ele iria depender dele próprio). Cada um dos parametros do Header tem um tamanho fixo, o que permite percorrer o Header de forma determinística aquando da escrita e da leitura: o nome do autor e a palavra passe ocupam ambos 10 bytes, o número de Assets é representado em apenas um byte e o nonce ocupa 1 byte. Estes foram os valores escolhidos pois considerámos que 10 bytes seriam necessários para guardar um nickname ou uma password e um byte seria necessário para guardar o número de Assets, permitindo ao utilizador um total de 256 Assets por TPDV.

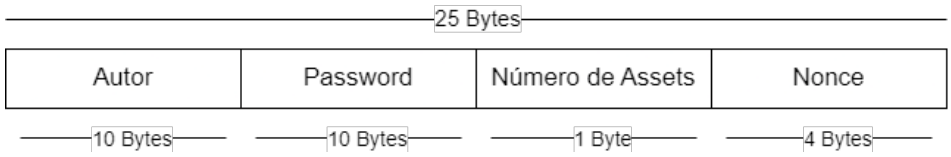

Figura 1: Header do TPDV

O utilizador fornece os dados necessários à criação do TPDV (o seu nome, a sua palavra passe e o nome do TPDV) e o ficheiro é criado caso não exista. Uma vez tendo estes dados é criado o Header com 0 no número de Assets e o Nonce vazio. O Header (unsigned char *header) é enviado para o Enclave por uma ecall. Uma ecall é um método seguro para enviar dados para o enclave de forma protegida e confidencial. Estes dados vão então ser selados com a chave do próprio Enclave. O "seal"de dados no contexto do Intel SGX é um processo seguro usado para proteger e cifrar os dados antes de serem armazenados fora de um enclave. Isso garante a confidencialidade e integridade dos dados, permitindo que apenas o enclave autorizado possa acessá-los posteriormente. Por fim, o ecall devolve um array com os dados selados que vão ser escritos no ficheiro que o utilizador indicou previamente. Foi tida em consideração que o tamanho dos dados selados é significativamente maior ao tamanho dos dados *raw* ao longo do projeto para alocar memória de forma eficiente.

2.2 Adicionar um Asset

Para adicionar um Asset, o utilizador fornece primeiramente o nome do TPDV ao qual quer adicionar o dito Asset, com o seu nome de autor e palavra passe. Caso o TPDV não exista ou o nome e palavra passe não correspondam ao que está escrito no TPDV, a operação de adicionar um Asset é cancelada de imediato. Estas verificações são recorrentes em todas as funcionalidades para não comprometer os dados do autor original. Estas verificações implicam que o TPDV seja lido pela aplicação insegura, que os dados lidos sejam enviados para o Enclave via ecall, que seja feito o "unseal"destes mesmos dados e, que, por fim, sejam comparados os dados fornecidos com os dados lidos, autenticando ou não o utilizador.

Os Assets vão ser adicionados no final de cada TPDV, não sendo possível a inserção noutro local. A estrutura do TPDV todo é representada pela figura 2. Cada Asset é representado por 3 campos: o nome do ficheiro do Asset (20 bytes), tamanho do Asset (4 bytes) e o Asset propriamente dito (x bytes). Desta forma é também possível ler os Assets sem necessitar de um separador, pois para avançar um Asset saltam-se os 20 bytes do nome, os 4 bytes do tamanho do Asset e salta-se o tamanho do Asset em questão.

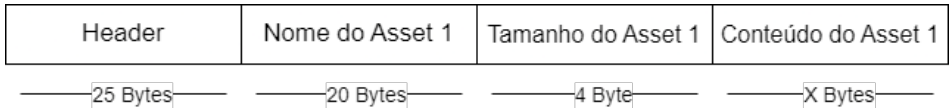

Figura 2: Estrutura do TPDV

O utilizador fornece o nome do ficheiro que quer adicionar e a sua informação é lida caso ele exista. Uma vez tendo os dados selados do TPDV e os dados raw do Asset, calculou-se o tamanho final do conjunto selado TPDV+Asset com a seguinte fórmula:

$$New\_TPDV\_size = Sealed\_sizeof(Unsealed\_sizeof(TPDV) + sizeof(Asset) + 20 + 4) \quad (1)$$

Assim, é possível alocar o tamanho exato de memória que os dados selados do TPDV com o Asset adicionado vai ocupar. É então feito um ecall que leva os dados do TPDV selados e os dados do Asset para o Enclave fazer a sua junção, devolvendo o conteudo selado para a App escrever no ficheiro. É nesta fase que é feita a verificação de integridade (Figura 3). A verificação de integridade é algo feito antes de qualquer uma das funcionalidades descritas (exceto a criação de TPDV). Após o unseal dos dados, o Encalve calcula o hash dos Assets do TPDV (caso existam) e compara-os com o nonce guardado no header. Em caso de diferença, o Enclave devolve um erro a dizer que a integridade do TPDV foi comprometida.

```
int check_nonce(unsigned char *nonce, unsigned char *content) {
    unsigned char new_nonce[4] = {0};
    calculate_nonce(content, new_nonce);
    if (memcmp(nonce, new_nonce, 4) != 0) {
        return 0;
    }
    return 1;
}
```

Figura 3: Verificação do Nonce

O Asset é então adicionado após o Header caso seja o primeiro Asset do TDPV, caso contrário, é obtido o número total de Assets no TPDV (através do Header), e percorrem-se todos eles para adicionar o

novo Asset no final do TPDV, criando então uma chain de Assets onde a inserção só é possível no final.

Uma vez tendo adicionado o Asset, é necessário alterar o nonce consoante o conjunto dos dados novos do TPDV. É então calculado o hash do conjunto dos Assets e o nonce é alterado para os últimos 4 bytes do hash.

Em caso de sucesso, o resultado da adição é o seguinte:

```
Introduza o nome do ficheiro TPDV: mytpdv
Introduza o nome do autor: fabio
Introduza uma palavra passe: password
Introduza o nome do asset: Draft.pdf

ENCLAVE: Adding asset: Draft.pdf
```

Figura 4: Adição de um Asset ao TPDV

2.3 Listar os Assets de um TPDV

Para listar os Assets de um TPDV, o utilizador fornece o nome do TPDV, o seu nome de autor e a palavra passe. Caso o TPDV não exista ou o nome e palavra passe não correspondam ao que está escrito no TPDV, a operação de listar os Assets é cancelada de imediato. Estas verificações são recorrentes em todas as funcionalidades para não comprometer os dados do autor original. Estas verificações implicam que o TPDV seja lido pela aplicação insegura, que os dados lidos sejam enviados para o Enclave por meio de um ecall, que seja feito o "unseal"destes mesmos dados e que, por fim, sejam comparados os dados fornecidos com os dados lidos, autenticando ou não o utilizador.

O Enclave lê os dados após o unseal e imprime o conteúdo do TPDV. Uma vez que o tamanho do cabeçalho é fixo, o acesso às variáveis que ele possui é trivial. Basta aceder ao ponteiro que contém os dados e adicionar-lhe o offset do tamanho da variável em questão, como mostra a Figura 5.

```
unsigned char actual_author[AUTHOR_SIZE] = {0};
unsigned char actual_password[PW_SIZE] = {0};
memcpy(actual_author, temp_buf, AUTHOR_SIZE);
memcpy(actual_password, temp_buf + AUTHOR_SIZE, PW_SIZE);
unsigned char num_assets = temp_buf[AUTHOR_SIZE + PW_SIZE];
```

Figura 5: Acesso às variáveis do Header

Para imprimir os Assets propriamente ditos o processo é ligeiramente diferente. Uma vez que o tamanho de cada Asset é diferente, deve ter-se isso em consideração. Assim, usou-se um ponteiro que percorria os dados e imprimia-os. Após se ignorar o Header, os primeiros 20 bytes de cada Asset correspondem ao tamanho do nome do Asset, os 4 seguintes são onde é guardado o tamanho do Asset e os próximos bytes são o próprio Asset. Uma vez que o tamanho do Asset ocupa 4 bytes, é necessário concatenar o valor numa variavel única. Esta operação foi feita por Bitwise ORs sucessivos para guardar o tamanho do Asset num uint32_t. A Figura 6 mostra o processo de impressão dos Assets.

```
uint32_t pointer = HEADER_SIZE; // Skip the header
for (int i = 0; i < num_assets; i++) {
    unsigned char asset_name[20] = {0};
    unsigned char asset_size_bytes[4] = {0};
    uint32_t asset_size = 0;

    // Get the asset name
    memcpy(asset_name, temp_buf + pointer, 20);
    printf("ENCLAVE: Asset name: %s\n", asset_name);

    pointer += 20;
    asset_size |= (uint32_t)temp_buf[pointer++] << 24;
    asset_size |= (uint32_t)temp_buf[pointer++] << 16;
    asset_size |= (uint32_t)temp_buf[pointer++] << 8;
    asset_size |= (uint32_t)temp_buf[pointer++];
    printf("ENCLAVE: Asset size: %d\n", asset_size);

    unsigned char asset_content[asset_size];
    memcpy(asset_content, temp_buf + pointer, asset_size);
    printf("ENCLAVE: Asset content: %s\n", asset_content);
    pointer += asset_size;
    printf("\n");
}
```

Figura 6: Loop de Impressão dos Assets

O resultado é mostrado da Figura 7:

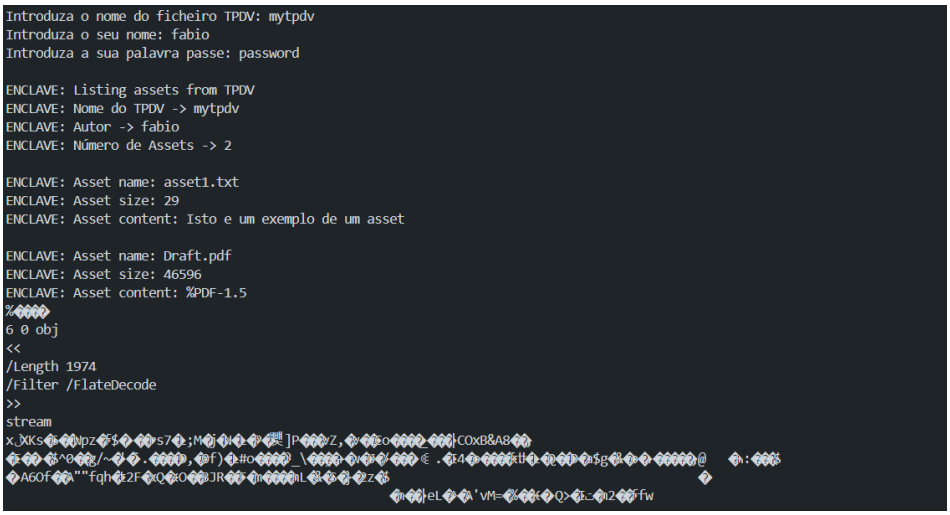

Figura 7: Resultado da listagem

2.4 Extrair um Asset de um TPDV

O processo de extrair um Asset é semelhante ao início das opções anteriores. O lado inseguro da aplicação lê os dados selados guardados no ficheiro e envia-os para o Enclave. O Enclave recebe os dados e o índice do Asset que vai ser extraído. Para isso, o utilizador deve listar previamente os Assets para saber o índice do Asset que pretende extrair. O Enclave verifica se o Asset existe, ou seja, se o índice fornecido é menor ou igual ao número de Assets presente no cabeçalho e prossegue para a sua obtenção. É usado um ciclo semelhante ao da Figura 6, porém, desta vez, em vez da condição de paragem ser o último Asset, o ponteiro para imediatamente antes do Asset correspondente ao índice fornecido. O nome do Asset e o seu conteúdo são enviados às claras para a aplicação insegura para esta criar um ficheiro para esse Asset na pasta "./Extractions/".

2.5 Comparar um hash com um hash de um Asset do TPDV

É também possível fornecer ao programa um hash para comparar com o hash de um Asset presente num certo TPDV. Isto permite ao utilizador perceber se a integridade dos seus Assets não foi comprometida. Assim, o utilizador fornece o sha256 do seu ficheiro local e o índice ao qual ele corresponde no TPDV. Usando funções próprias do SGX para o efeito, calculou-se o SHA256 do Asset selecionado pelo utilizador e compara-se com o resultado da digest function calculado anteriormente. Em caso de sucesso, o Enclave afirma que os Hashes correspondem, como mostra a Figure 8. Na Figura 9 é observa-se o hash do Asset selecionado fornecido pelo OpenSSL.

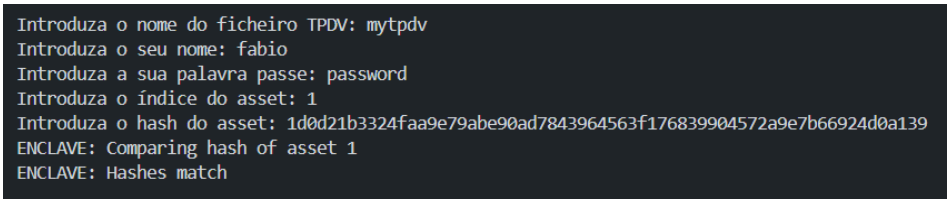

Figura 8: Resultado da comparação dos Hashes

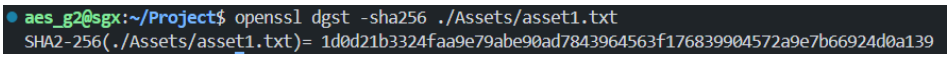

Figura 9: Resultado do Hash pelo OpenSSL

2.6 Clonar o conteúdo do TPDV

A última funcionalidade implementada é o clone do conteúdo do TPDV. Esta funcionalidade necessita de um segundo Enclave, uma vez que ela envolve a clonagem dos dados selados por um Enclave para outro Enclave. O processo passa por fazer unseal dos dados de um Enclave e enviá-los para o segundo Enclave para este selar com a sua chave. Esta é uma alternativa pouco segura, uma vez que os dados ficam expostos durante a passagem de um Enclave para outro, através da aplicação não segura. Para contornar essa limitação, os dados são cifrados com uma chave partilhada entre os dois Enclaves, mediante uma troca de chaves Diffie-Helman. O procedimento completo é o seguinte:

1. Fazer uma troca de chaves Diffie-Helman entre os dois Enclaves para obter uma shared key.
2. Fazer unseal dos dados do primeiro Enclave e cifrá-los com a shared key.
3. Enviar os dados cifrados (diferente de selados) do primeiro Enclave para a App.
4. Enviar os dados cifrados da App para o segundo Enclave.
5. Decifrar os dados com a shared key e fazer o seal com a chave do segundo Enclave.
6. Enviar os dados selados pelo segundo Enclave para a App.
7. Guardar os dados selados num novo ficheiro (clone).

A Figura 10 mostra um exemplo de chaves partilhadas pelos dois enclaves, após as comunicações do protocolo Diffie-Helman.

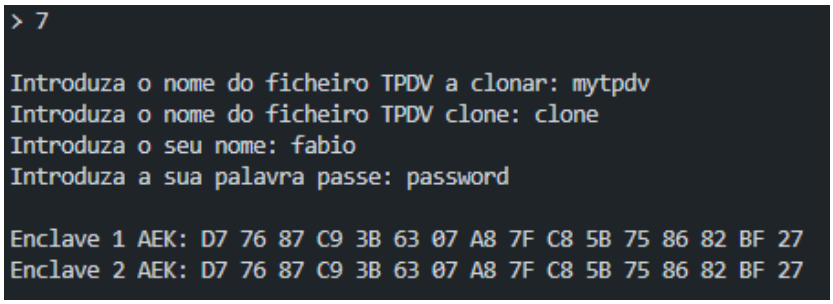

Figura 10: Chaves partilhadas entre os Enclaves

A partir deste momento, passaram a existir 2 Enclaves, pelo que se o utilizador pretender listar o conteúdo de um TPDV, por exemplo, precisa de identificar a qual Enclave o TPDV pertence, pois os Enclaves tem chaves diferentes e um não consegue fazer o unseal de dados selados por outro. Apenas foi acionada a opção de escolher o Enclave no processo de listagem do Assets, para demonstrar que o clone foi feito com sucesso. A Figura 11 mostra a tentativa de leitura de um ficheiro clonado a partir dos dois Enclaves. É possível observar que o 1 Enclave é incapaz de ler os dados selados pelo segundo Enclave, enquanto o segundo Enclave mostra os dados originais.

```
> 3

Introduza o nome do ficheiro TPDV: clone
Introduza o Enclave onde está o TPDV (1 ou 2): 1
Introduza o seu nome: fabio
Introduza a sua palavra passe: password

ENCLAVE: Listing assets from TPDV
ENCLAVE: Error unsealing data

Selecione uma opção
1 - Criar um novo TPDV file
2 - Adicionar um asset (ficheiro binário) ao TPDV
3 - Listar todos os assets guardados num TPDV file
4 - Extrair um asset
5 - Comparar um hash com o hash de um asset presente num TPDV
6 - Mudar a password de um TPDV
7 - Clonar o conteudo do TPDV
0 - Sair
> 3

Introduza o nome do ficheiro TPDV: clone
Introduza o Enclave onde está o TPDV (1 ou 2): 2
Introduza o seu nome: fabio
Introduza a sua palavra passe: password

ENCLAVE: Listing assets from TPDV
ENCLAVE: Nome do TPDV -> clone
ENCLAVE: Autor -> fabio
ENCLAVE: Número de Assets -> 3

ENCLAVE: Asset name: asset1.txt
ENCLAVE: Asset size: 29
ENCLAVE: Asset content: Isto e um exemplo de um asset
```

Figura 11: Leitura dos dados clonados

3 Conclusão

O uso do Intel SGX neste projeto demonstrou os benefícios da utilização de Enclaves para proteger dados sensíveis. Ao implementar o TPDV com base nessa tecnologia, aprendemos a projetar sistemas que oferecem um alto nível de segurança, mantendo a flexibilidade necessária para operações complexas de gestão de dados.

A capacidade de criar, adicionar, listar, extrair e verificar ativos num ambiente protegido pelo Intel SGX garante que informações confidenciais possam ser manipuladas com segurança, sem comprometer a privacidade dos utilizadores. A clonagem segura entre enclaves também abre possibilidades para cenários avançados de partilha de dados entre ambientes isolados.

Em resumo, o uso do Intel SGX neste projeto representa um passo importante na implementação de soluções que combinam segurança robusta com funcionalidades avançadas de gestão de dados, proporcionando uma experiência confiável e protegida para os utilizadores finais.