

Passo a Passo Projeto SBRC

Francisco Thiago dos Santos Gonçalves

December 2023

1 Introdução

Esse documento consiste em um passo a passo para reproduzir os experimentos realizados no trabalho [1]. O projeto se baseia na proposta de ofuscamento apresentado no trabalho realizado por [2] e utiliza métodos de otimização Heurísticos para reduzir o overhead causado pelo ofuscamento dos pacotes.

2 Baixando o Projeto

Inicialmente, é necessário baixar os scripts e datasets salvos no github. Para isso, basta utilizar o seguinte comando.

```
git lfs clone https://github.com/franciscothiagoufc/Uma-Heuristica-para-Ofuscacao-de-Trafego-em-Internet-das-Coisas
```

Vale ressaltar que é necessário possuir a ferramenta git e a extensão git lfs para executar o comando. Após isso, uma pasta nomeada “**Uma-Heuristica-para-Ofuscacao-de-Trafego-em-Internet-das-Coisas**” será salva no diretório onde o comando foi executado. Dentro desse diretório, na pasta Data/Raw, são salvos arquivos compactados contendo os datasets utilizados. Basta utilizar alguma ferramenta para descomprimir-los.

3 Utilizando Método Heurístico

A fim de minimizar o Overhead, o trabalho propõe utilizar dados estatísticos do envio do pacote e aplicar métodos de otimização para definir o tamanho do pacote após o padding. Na pasta **src/Optimization**, o script **distribuiton.py** lê os datasets e calcula a frequência relativa de envio de pacotes com tamanho entre 1 e 1500 bytes e salva esses valores no arquivo **distribution.txt**. Para executar o script, basta entrar em **src/Optimization** e utilizar o comando

```
python3 distribution.py
```

Ainda nessa pasta, o arquivo **optimizedPaddingValues.py** utiliza essa distribuição para obter o tamanho do preenchimento que minimize o overhead.

Após o preenchimento, o tamanho dos pacotes é alterado para um conjunto discreto de valores ¹, o tamanho desse conjunto é definido pelo variável M na linha 4 do script².

Por exemplo, para M=5, os valores obtidos no projeto foram 66, 123, 235, 543 e 1500. O valor de M define a quantidade de frames utilizados, como é mencionada no artigo. Para executar esse script, basta utilizar o comando

```
python3 optimizedPaddingValues.py
```

Ainda no cenário em que M=5, o script irá gerar um arquivo chamado padding5.txt. Ele contém um dicionário relacionando o tamanho original do pacote antes do preenchimento e o novo tamanho após o preenchimento. Esses arquivos são utilizados nas etapas seguintes. Vale ressaltar que a mesma lógica pode ser utilizada para outros valores de M.

Ademais, vale ressaltar que essas etapas já foram realizadas para M=2, 3, 4 e 5.

4 Aplicando Padding e Testando Proposta

Uma vez que os valores do preenchimento são obtidos de forma heurística, o método de preenchimento pode ser aplicado nos pacotes contidos nos datasets. Na pasta src, execute o script **createsFolderStructure.py** para que parte da estrutura dos diretórios seja criada.

Na pasta **Data/Processed**, crie o diretório **OptimizedPadding**. Nessa pasta, serão salvos os pacotes após o preenchimento ser aplicado. Para escolher o nível de preenchimento utilizado (quantidade de frames utilizadas), entre no arquivo **experimentConfiguration.json** na pasta **Data/Configuration** e altere os campos **paddingStrategy** e **frames** para o valor desejado.

Por exemplo, caso o teste utilize 3 frames apenas, esses dois campos devem possuir o valor 3. Vale ressaltar que, nesse exemplo, é necessário que exista o arquivo **padding3.txt** na pasta **src/Optimization**. A construção desses arquivos é explicada na sessão anterior.

Na pasta **Data/OptimizedPadding** crie uma pasta nomeada com a quantidade de frames que será utilizada (eg: **src/Optimization/3**). Na pasta **src/** execute o script **optimizedPaddingValues.py**

```
python3 optimizedPaddingValues.py
```

Os pacotes já preenchidos serão salvos na pasta criada anteriormente. Ainda na pasta **src** execute o script **prepare_features.py**.

```
python3 prepare_features.py
```

¹No artigo, esses valores são referidos como tamanho dos frames

²M é a quantidade de frames utilizados

Esse script prepara os datasets para os algoritmos de machine learning utilizados. Por fim, execute o script `padding_strategies_evaluation.py` para analisar a eficácia dos métodos de preenchimento em esconder os fingerprints presentes no tamanho dos pacotes enviados.

```
python3 prepare_features.py
```

Dois testes diferentes podem ser feitos: Utilizando train test split e o cross validation. Para executar ou não cada teste, basta manipular as funções nas linhas entre 162 e 165. Os resultados são salvos como arquivos txt na pasta src.

References

- [1] Francisco Gonçalves, Antonio Pinheiro, Críston Souza, and Jeandro Bezerra. Uma heurística para ofuscação de tráfego em internet das coisas. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 449–462, Porto Alegre, RS, Brasil, 2023. SBC.
- [2] Antônio J. Pinheiro, Paulo Freitas de Araujo-Filho, Jeandro de M. Bezerra, and Divanilson R. Campelo. Adaptive packet padding approach for smart home networks: A tradeoff between privacy and performance. *IEEE Internet of Things Journal*, 8(5):3930–3938, 2021.