

Artefato SBSeg'25 Apêndice: Artigo #[Seu ID de Submissão Aqui]

Walter H.A. Santos¹, Kauã O. S. Menezes¹, Lucas W. Santos¹, Marcos V. S. Morais¹

¹ Departamento de Sistemas de Informação¹
Universidade Federal de Sergipe (UFS) – Itabaiana, SE – Brazil

{ waltersantos1, okaua057, lucas.werthein, imarcossm}@academico.ufs.br

***Resumo.** Este artigo apresenta uma visualização interativa do protocolo de distribuição de chaves quânticas BB84, implementada com Python, Qiskit e Streamlit. O objetivo é fornecer uma ferramenta didática para o ensino de criptografia quântica, permitindo a simulação do protocolo, incluindo a presença de um espião, e a observação do impacto na Taxa de Erro Quântico (QBER). O artefato associado permite a replicação dos resultados experimentais apresentados, contribuindo para a compreensão prática da segurança em QKD.*

O apêndice permite que os autores forneçam mais detalhes para os revisores sobre o artefato. Utilize este documento para passar informações de acordo com os selos requisitados. Caso seja necessário acesso a recursos externos (ex: hardware específico, APIs pagas, dados confidenciais), o respectivo processo e credenciais devem estar descritos neste documento. Neste caso, não há tais requisitos.

1. Selos Considerados

Solicitamos a avaliação deste artefato para a obtenção dos seguintes selos de avaliação da ACM, conforme definidos nas diretrizes do SBSeg'25:

- **Artifact Available (Disponível):** O artefato está publicamente acessível no repositório GitHub (link na Seção 2) sob uma licença MIT.
- **Artifact Functional (Funcional):** O artefato funciona conforme descrito. As instruções de instalação (Seção 4) e o teste mínimo (Seção 5) demonstram a operacionalidade da aplicação interativa principal.
- **Artifact Reusable (Reutilizável):** O código é modular (separação entre lógica do BB84 e interface), utiliza bibliotecas Python padrão, está documentado (README, comentários) e possui licença permissiva (MIT), facilitando a reutilização e extensão por terceiros.
- **Results Reproduced (Resultados Reproduzidos):** O artefato permite a reprodução dos resultados chave do artigo (Tabela 1). A Seção 6 descreve como proceder, referenciando o arquivo ARTEFATO.md no repositório para os passos detalhados.

Portanto, os selos considerados para avaliação são: **Available**, **Functional**, **Reusable** e **Results Reproduced**.

2. Informações Básicas e de Acesso

O código fonte completo, scripts, arquivos de dependência e documentação associados a este artefato podem ser encontrados no seguinte repositório público do GitHub:

`https://github.com/WalterHenri/ImplBB84`

O repositório contém:

- `README.md`: Instruções gerais de configuração, visão geral e requisitos.
- `LICENSE`: Arquivo contendo a licença MIT.
- `requirements.txt`: Arquivo para instalação das dependências Python via `pip`.
- Arquivos `.py`: Código fonte da aplicação (lógica do BB84, interface Streamlit, utilitários).

Não são necessárias credenciais de acesso, APIs pagas, hardware especializado ou qualquer software proprietário para utilizar este artefato. Todo o software necessário é de código aberto e de fácil obtenção.

Estimamos que o tempo total para clonar o repositório, instalar as dependências e executar o teste mínimo (Seção 5) seja inferior a 5 minutos em uma conexão de internet razoável. A reprodução completa dos resultados da Tabela 1 (Seção 6) deve levar aproximadamente **5-10 minutos** (dependendo da máquina utilizada).

3. Dependências

As dependências de software necessárias para executar o artefato são:

- **Python**: Versão 3.13.x.
- **Pip**: O gerenciador de pacotes do Python, geralmente incluído na instalação padrão do Python.
- **Bibliotecas Python**: As bibliotecas específicas e suas versões exatas estão listadas no arquivo `requirements.txt`. As principais incluem:
 - `qiskit`: Framework principal para computação e simulação quântica.
 - `qiskit-aer`: Componente do Qiskit para simulação de alto desempenho.
 - `streamlit`: Framework para criação da interface web interativa.
 - `numpy`: Para operações numéricas e manipulação de arrays.
 - `matplotlib`: Usada implicitamente para alguns gráficos.

Importante: A instalação se restringe ao ambiente Python e suas bibliotecas via `pip`.

4. Instalação

O processo de instalação é padrão para um projeto Python e deve ser realizado em um terminal ou linha de comando:

1. **Clonar o Repositório**: Obtenha o código fonte do GitHub:

```
git clone https://github.com/WalterHenri/ImplBB84.git
```

2. **Navegar para o Diretório**: Entre na pasta do projeto clonado:

```
cd ImplBB84
```

3. **Instalar Dependências:** Instale todas as bibliotecas Python necessárias listadas em `requirements.txt`:

```
pip install -r requirements.txt
```

Após estes passos, o ambiente estará pronto para executar a aplicação.

5. Teste Mínimo de Funcionalidade

Para verificar se a instalação foi bem-sucedida e se o artefato está funcional em seu nível básico, execute o seguinte comando no terminal (com o ambiente virtual ativado):

```
streamlit run app.py
```

Resultado Esperado: Este comando deve iniciar o servidor Streamlit e abrir automaticamente a interface da aplicação "Visualização Interativa do Protocolo BB84" no seu navegador web padrão. A interface deve carregar sem erros, exibindo os controles para definir parâmetros (número de qubits, presença de espião) e as áreas para visualização dos resultados da simulação.

A execução bem-sucedida deste teste demonstra que as dependências foram instaladas corretamente e que a aplicação principal é capaz de iniciar, validando o selo "Artifact Functional".

6. Execução dos Experimentos e Reprodução de Resultados

Este artefato foi projetado para permitir a reprodução dos resultados quantitativos apresentados na **Tabela 1** do artigo associado (valores médios de QBER e tamanho da chave final para os cenários com e sem a presença de um espião 'Eve').

Além da reprodução dos resultados específicos do artigo, a aplicação interativa (`streamlit run app.py`) pode ser usada para exploração livre e didática do protocolo BB84, alterando parâmetros e observando os efeitos em tempo real, o que contribui para a reutilização do artefato (selo "Artifact Reusable").