Instituto Tecnológico de Aeronáutica

CES-35 — Redes de Computadores

Exame: FANET

Professora:
Cecília de Azevedo Castro César

Alunos:
Denys Derlian Carvalho Brito
Lucas Silva Lima
Rafael Hoffmann Giannico

11 de dezembro de 2024

Conteúdo

1 Introdução **2**

2 Objetivos **2**

3 Cenários e Metodologia **3**
3.1 Cenários Simulados . . . 3
3.2 Metodologia . . . 3
3.2.1 Desenvolvimento do Ambiente de Simulação . . . 3
3.2.2 Configuração dos Componentes . . . 4
3.2.3 Implementação da Comunicação e Decisão . . . 4
3.2.4 Simulação dos Cenários . . . 4

4 Estrutura e Funcionamento da Solução **5**
4.1 Arquitetura do Sistema . . . 5
4.2 Módulos Implementados . . . 5
4.2.1 `main.py` . . . 5
4.2.2 `adhoc.py` . . . 5
4.2.3 `drone.py` . . . 5
4.2.4 `baseStationControl.py` . . . 6
4.2.5 `hacker.py` . . . 6
4.2.6 `encryption.py` . . . 6
4.2.7 `message.py` . . . 6
4.2.8 `globals.py` . . . 6
4.2.9 `uav.py` . . . 6
4.3 Comunicação e Propagação de Mensagens . . . 7
4.4 Decisão Descentralizada . . . 7
4.5 Segurança e Criptografia . . . 7
4.6 Execução de Missões . . . 8
4.7 Resiliência a Ataques . . . 8

5 Testes e Resultados **8**
5.1 Funcionamento Cíclico do Sistema . . . 8
5.2 Funcionamento com Ataque de Replay . . . 8
5.3 Funcionamento com Contramedidas Ativas . . . 9
5.4 Análise Comparativa dos Resultados . . . 9

6 Conclusão e Trabalhos Futuros **9**

Denys Derlian
Lucas Lima
Rafael Hoffmann

1 Introdução

As Redes Ad Hoc Voadoras (FANETs - *Flying Ad Hoc Networks*) são fundamentais para aplicações críticas, como busca e resgate, monitoramento ambiental e entregas em áreas remotas. Projetadas para veículos aéreos não tripulados (UAVs), essas redes se caracterizam por comunicação distribuída, sem infraestrutura fixa, proporcionando flexibilidade em cenários dinâmicos.

Este projeto explora as capacidades de uma FANET em um ambiente simulado, com foco em como a rede se comporta sob ataques de *replay*, onde mensagens interceptadas são retransmitidas maliciosamente. A ausência de infraestrutura fixa torna as FANETs vulneráveis a ataques que podem comprometer missões sensíveis, destacando a necessidade de estratégias de mitigação.

O estudo investiga conceitos como:

- Comunicação restrita entre drones vizinhos.
- Decisão descentralizada baseada em informações locais.
- Proteção por criptografia simétrica e identificadores únicos (*nonce*).

Cenários simulados incluem operações em áreas remotas, regiões urbanas densas e situações de desastre, analisando a resiliência da rede com e sem contramedidas de segurança.

2 Objetivos

O projeto tem como objetivo geral desenvolver e simular uma Rede Ad Hoc Voadora (FANET) composta por drones, avaliando sua segurança e eficiência em cenários críticos. Além disso, busca-se identificar vulnerabilidades na comunicação distribuída e propor soluções para mitigar ataques cibernéticos.

Os objetivos específicos incluem:

- Desenvolver uma FANET funcional para missões dinâmicas.
- Implementar um mecanismo de decisão descentralizada com comunicação restrita.
- Integrar segurança na comunicação por meio de criptografia e *nonce*.
- Simular cenários representativos, como áreas remotas e regiões urbanas.
- Avaliar a resiliência a ataques de *replay*.

O projeto visa compreender as dinâmicas de uma FANET em condições adversas, propondo soluções escaláveis para aplicações práticas.

3 Cenários e Metodologia

Nesta seção, são apresentados os cenários utilizados na simulação e a metodologia adotada para implementar e avaliar o desempenho da FANET proposta. A escolha dos cenários reflete aplicações reais de redes ad hoc voadoras, enquanto a metodologia detalha as etapas de desenvolvimento, configuração e experimentação.

3.1 Cenários Simulados

Para avaliar a solução desenvolvida, foram projetados três cenários distintos, representando diferentes condições e desafios operacionais:

- Desastres Naturais
- Regiões de Difícil Acesso
- Áreas Urbanas Densas

3.2 Metodologia

A metodologia adotada foi planejada para garantir uma análise abrangente e confiável do comportamento da FANET em diferentes condições. Os passos seguidos incluem:

3.2.1 Desenvolvimento do Ambiente de Simulação

O ambiente de simulação foi desenvolvido utilizando a linguagem Python, com o suporte das seguintes ferramentas:

- **Pygame**: Para renderização gráfica da simulação, permitindo visualização das posições e interações entre os drones.
- **NumPy**: Para cálculos matemáticos e manipulação de vetores tridimensionais, como distância entre drones e pontos de interesse.
- **Cryptography**: Para implementação de mecanismos de segurança, incluindo criptografia simétrica e controle de *nonce*.

A FANET foi modelada como uma rede distribuída, onde cada drone opera como uma instância autônoma, conectada aos dois vizinhos mais próximos.

3.2.2 Configuração dos Componentes

Os seguintes componentes foram configurados no ambiente de simulação:

- **Estação Base (GCS)**: Responsável por iniciar missões e coordenar o envio de mensagens de descoberta (*DISCOVER*) para a rede.
- **Drones (UAVs)**: Nós móveis da FANET, equipados com lógica para tomada de decisão descentralizada e propagação de mensagens.
- **Hacker**: Simula o atacante que intercepta mensagens legítimas, armazena e retransmite pacotes (*Replay Attack*) para comprometer a operação da rede.

Cada drone foi configurado com uma posição inicial aleatória, velocidade constante e capacidade de se comunicar apenas com os dois drones mais próximos.

3.2.3 Implementação da Comunicação e Decisão

O protocolo de comunicação foi projetado com os seguintes tipos de mensagens:

- **DISCOVER**: Enviada pela estação de solo para identificar o drone mais próximo de um ponto de interesse.
- **RETURN**: Drones retornam à estação a informação sobre qual deles está mais próximo.
- **EXECUTE**: Inicia a execução da missão pelo drone selecionado.
- **COMPLETE**: Informa que o ponto de interesse foi alcançado.
- **FINISH**: Finaliza a missão na estação base.

A lógica de decisão é descentralizada, baseada em informações locais dos drones, como distância ao ponto de interesse e mensagens propagadas pela rede.

3.2.4 Simulação dos Cenários

Cada cenário foi executado em condições específicas, avaliando o impacto dos seguintes fatores:

- Distância entre os drones e o ponto de interesse.
- Eficiência da propagação de mensagens na rede.
- Impacto de ataques de *replay* no comportamento da rede.

4 Estrutura e Funcionamento da Solução

A solução foi desenvolvida como uma simulação de uma Rede Ad Hoc Voadora (FANET), composta por drones, uma estação de solo e um atacante. Nesta seção, são detalhados os principais componentes e funcionalidades implementados, com exemplos de código que ilustram os conceitos abordados.

4.1 Arquitetura do Sistema

A arquitetura é composta pelos seguintes elementos principais:

- **Estação Base (GCS)**: Responsável por coordenar as missões e iniciar a comunicação com os drones.
- **Drones (UAVs)**: Nós móveis da rede ad hoc que realizam missões de forma descentralizada, propagando mensagens e tomando decisões localmente.
- **Atacante (Hacker)**: Simula um nó malicioso que captura e retransmite mensagens para comprometer a operação da rede.

4.2 Módulos Implementados

O código foi estruturado em vários módulos, cada um responsável por funcionalidades específicas da simulação da FANET. A seguir, são descritos os principais módulos e suas responsabilidades:

4.2.1 `main.py`

Este módulo é o ponto de entrada da aplicação. Ele coordena o ciclo de vida da simulação, inicializando os drones, a estação de solo e o atacante, além de gerenciar o fluxo de comunicação e a renderização gráfica.

4.2.2 `adhoc.py`

Este módulo implementa a lógica central da rede ad hoc, permitindo a comunicação entre os componentes (drones, estação de solo e atacante). Ele gerencia a propagação de mensagens e a conectividade entre os nós.

4.2.3 `drone.py`

Define o comportamento dos drones, incluindo sua capacidade de mover-se até o ponto de interesse, tomar decisões descentralizadas e interagir com outros drones.

Este módulo também implementa a lógica de processamento de mensagens e o gerenciamento de criptografia.

4.2.4 `baseStationControl.py`

Este módulo representa a estação de solo (GCS), que é responsável por iniciar as missões, enviar mensagens de descoberta (`DISCOVER`) e receber o retorno dos drones sobre o nó mais próximo do ponto de interesse. Ele também finaliza missões com a mensagem `FINISH`.

4.2.5 `hacker.py`

Modela o comportamento do atacante, que intercepta mensagens legítimas, armazena pacotes e tenta comprometê-los por meio de ataques de *replay*. A lógica de retransmissão de pacotes maliciosos está implementada neste módulo.

4.2.6 `encryption.py`

Este módulo implementa os mecanismos de criptografia e descriptografia utilizados para proteger as mensagens trocadas na rede. Ele utiliza criptografia simétrica baseada no algoritmo AES e incorpora *nonce* para evitar ataques de *replay*.

4.2.7 `message.py`

Este módulo define a estrutura das mensagens trocadas na rede, incluindo campos como `source_id`, `destination_id`, `mission_id` e `type`. Ele serve como base para a comunicação entre os componentes.

4.2.8 `globals.py`

Contém variáveis globais utilizadas em toda a aplicação, como dimensões da área simulada (`LARGURA` e `ALTURA`), cores para renderização gráfica (`BLACK`, `WHITE`, etc.) e configurações gerais.

4.2.9 `uav.py`

Define a classe base para os drones (`UAV`), implementando funcionalidades genéricas como posicionamento, velocidade e identificação. Este módulo é estendido por `drone.py` para adicionar funcionalidades específicas.

4.3 Comunicação e Propagação de Mensagens

A comunicação entre os componentes é baseada em mensagens distribuídas pela rede ad hoc. As mensagens implementadas incluem:

- **DISCOVER**: Inicia o processo de descoberta do drone mais próximo.
- **RETURN**: Envia à estação base o resultado da descoberta.
- **EXECUTE**: Inicia a execução de uma missão pelo drone selecionado.
- **COMPLETE**: Notifica a conclusão da missão.
- **FINISH**: Finaliza a missão na estação base.

4.4 Decisão Descentralizada

A decisão sobre qual drone deve realizar a missão é feita de forma distribuída, com base na distância ao ponto de interesse. O processo funciona assim:

- Quando uma mensagem `DISCOVER` chega, o drone calcula sua distância ao ponto de interesse.
- Se ele for o mais próximo até aquele momento, atualiza a mensagem com sua distância e seu identificador e a propaga para os dois drones vizinhos.
- Caso não seja o mais próximo, apenas encaminha a mensagem para os vizinhos, mantendo as informações do drone mais próximo registrado na mensagem.
- Esse processo continua até que todos os drones tenham recebido a mensagem e saibam quem é o mais próximo do ponto de interesse.

Se a decisão demorar muito (devido a atrasos na propagação ou problemas na rede), o sistema pode acabar escolhendo um drone que não seja o mais próximo, garantindo que a missão não fique bloqueada. Isso assegura que o ciclo de missões continue funcionando mesmo em condições adversas.

4.5 Segurança e Criptografia

A segurança das mensagens trocadas na FANET é garantida por meio de criptografia simétrica. Cada mensagem é protegida com uma chave simétrica compartilhada entre os drones e a estação de solo. Além disso, um vetor de inicialização (IV) é gerado aleatoriamente para cada mensagem, garantindo que mesmo mensagens idênticas resultem em criptogramas diferentes. Esse mecanismo protege a integridade e confidencialidade dos dados, dificultando ataques como interceptação ou *replay*.

4.6 Execução de Missões

Após a decisão descentralizada, o drone selecionado executa a missão, movendo-se até o ponto de interesse. O drone calcula a direção e começa a se mover em direção ao ponto alvo. Ao atingir o ponto alvo, ele executa a missão.

4.7 Resiliência a Ataques

A implementação de medidas de segurança permitiu mitigar ataques simulados. No ataque de *replay*, o atacante retransmite pacotes capturados para tentar enganar os drones. A proteção baseada em *nonce* e criptografia garantiu que mensagens inválidas fossem descartadas. Com este mecanismo, apenas mensagens da missão atual são processadas.

5 Testes e Resultados

Foram realizados três testes principais para avaliar o sistema:

5.1 Funcionamento Cíclico do Sistema

O sistema foi testado para verificar sua capacidade de operar continuamente, com novos pontos de interesse gerados após cada missão.

Resultados:

- Missões foram concluídas com sucesso e novas foram iniciadas sem interrupções.
- As mensagens `DISCOVER`, `EXECUTE`, `COMPLETE` e `FINISH` propagaram corretamente.
- Todos os drones participaram ativamente do ciclo.

5.2 Funcionamento com Ataque de Replay

Simulou-se um ataque onde um drone malicioso interceptava mensagens legítimas, armazenava-as e as retransmitia para se passar pela estação base, bloqueando o sistema.

Resultados:

- O atacante retransmitiu mensagens interceptadas, enganando o drone executante e recebendo a mensagem `COMPLETE`.

- O ciclo de missões foi bloqueado, impedindo a geração de novos pontos de interesse.
- Logs confirmaram que a rede ficou paralisada devido à falta de autenticação robusta das mensagens.

5.3 Funcionamento com Contramedidas Ativas

Com o uso de *nonce* e criptografia simétrica, o sistema foi protegido contra ataques de *replay*.

Resultados:

- Mensagens inválidas foram descartadas automaticamente.
- O sistema operou normalmente, completando missões e iniciando novas.
- Logs confirmaram que somente mensagens válidas foram processadas.

5.4 Análise Comparativa dos Resultados

Os três testes demonstraram o comportamento do sistema em diferentes condições:

- Em operação normal, o sistema manteve um funcionamento cíclico e eficiente.
- Sob ataque de *replay*, o sistema foi comprometido, destacando a vulnerabilidade da comunicação sem proteção.
- Com contramedidas ativas, o sistema foi resiliente ao ataque, validando a eficácia das soluções implementadas.

Esses resultados indicam que a implementação de *nonce* e criptografia é fundamental para garantir a segurança e a continuidade das operações em uma FANET.

6 Conclusão e Trabalhos Futuros

O projeto demonstrou a viabilidade de uma FANET segura e resiliente em cenários críticos. As contramedidas desenvolvidas se mostraram eficazes na proteção contra ataques de replay.

Como trabalhos futuros, sugere-se:

- Melhorar o algoritmo de decisão descentralizada para reduzir erros de escolha.
- Introduzir novos cenários de ataque e defesas avançadas.
- Expandir a simulação para redes maiores com mais drones e hackers.