

Alunos: Pedro Henrique Tenorio de Magalhães de Oliveira
Petersson Matos Cardoso Santana

Segurança em IoT com BitDogLab

Este trabalho apresenta a implementação e a avaliação de técnicas de comunicação e segurança em dispositivos IoT utilizando o Raspberry Pi Pico W da BitDogLab em conjunto com o protocolo MQTT, com foco em aplicações educacionais e ambientes escolares. São abordadas desde as etapas iniciais de configuração de rede Wi-Fi e conexão com brokers MQTT até mecanismos de autenticação, publicação segura de dados e proteção contra ataques de replay. Também é explorada a aplicação de criptografia leve para ofuscação de mensagens e a análise da escalabilidade dessas soluções em cenários com múltiplos dispositivos BitDogLab integrados em uma rede escolar.

Os códigos desenvolvidos estão no Git:

<https://github.com/EmbarcaTech-2025/tarefa-iot-security-lab-pedro-pet.git>

Objetivo

Configurar uma comunicação MQTT básica via Wi-Fi utilizando a BitDogLab com Raspberry Pi Pico W em C/C++ (SDK do Pico), aplicar autenticação no broker, implementar criptografia leve e proteger contra ataques de sniffing e replay.

Etapa 1: Conectando a BitDogLab ao Wi-Fi

Para realizar a conexão de rede via Wi-Fi utilizando o SDK Pico W em conjunto com o lwIP, o código fornecido foi executado após a remoção das etapas de transmissão MQTT e codificação, mantendo apenas os procedimentos de conexão e configuração de rede. A compilação ocorreu sem erros, conforme pode ser observado na imagem do console apresentada abaixo.

```
ninja: Entering directory `C:\Users\gasor\Documents\GitHub\tarefa-iot-security-lab-pedro-pet/build`  
[196/196] Linking CXX executable iot_security_lab.elf  
* Terminal will be reused by tasks, press any key to close it.
```

A rede local foi criada a partir do hotspot móvel do computador, permitindo a conexão do Raspberry Pi Pico W. A confirmação da conexão foi feita ao verificar a presença do dispositivo na lista de aparelhos conectados do hotspot, como mostrado na imagem a seguir.

Dispositivos conectados: 1 de 8		
Nome do dispositivo	Endereço IP	Endereço físico (MAC)
PicoW	192.168.137.252	28:cd:c1:11:c7:f7

Etapa 2: Setup MQTT básico para IoT

Após a conexão à rede local, os trechos do código referentes à transmissão MQTT foram reabilitados e o endereço IP do broker foi configurado. O objetivo foi conectar o dispositivo embarcado a um broker MQTT e publicar dados em tópicos específicos, com tratamento básico de erros. A conexão foi estabelecida utilizando a porta 1884, já que a porta padrão (1883) apresentava conflitos, provavelmente por estar em uso por outros programas. Também foram adicionadas regras no firewall do Windows para permitir a comunicação na nova porta.

```
C:\Users\gasor>mosquitto -c "C:\mosquitto_conf\mosquitto.conf" -v
1749061089: mosquitto version 2.0.21 starting
1749061089: Config loaded from C:\mosquitto_conf\mosquitto.conf.
1749061089: Opening ipv6 listen socket on port 1884.
1749061089: Opening ipv4 listen socket on port 1884.
1749061089: mosquitto version 2.0.21 running
```

Etapa 3: Publicação MQTT sem segurança

Na publicação sem autenticação, a BitDogLab foi configurada como publisher, enviando a mensagem "26.5" ao tópico "escola/sala1/temperatura". O arquivo de configuração do Mosquitto (mosquitto.conf) foi ajustado para permitir conexões anônimas, ativando a opção `allow_anonymous true`. A imagem abaixo demonstra o recebimento e retransmissão da mensagem pelo broker, bem como o recebimento pelos assinantes conectados.

```
1749062749: Received PUBLISH from bitdog1 (d0, q0, r0, m0, 'escola/sala1/temperatura', ... (4 bytes))
1749062749: Sending PUBLISH to auto-55966E35-118C-6457-BFCA-FA4D88378B5F (d0, q0, r0, m0, 'escola/sala1/temperatura', ..
C:\Users\gasor>mosquitto_sub -h localhost -p 1884 -t "escola/sala1/temperatura" -u "aluno" -P "senha123"
26.5
26.5
```

Etapa 4: Autenticação básica no Mosquitto

Para esta etapa, o arquivo de configuração do broker foi alterado para restringir o acesso apenas a clientes autenticados. A senha foi gerada conforme o procedimento do roteiro, resultando em um arquivo de senhas criptografadas. Com a nova configuração, a transferência de informações passou a ser protegida por autenticação e foi realizada com sucesso, como ilustrado na imagem abaixo.

```
C:\Users\gasor>mosquitto -c "C:\mosquitto_conf\mosquitto.conf" -v
1749064495: mosquitto version 2.0.21 starting
1749064495: Config loaded from C:\mosquitto_conf\mosquitto.conf.
1749064495: Opening ipv6 listen socket on port 1884.
1749064495: Opening ipv4 listen socket on port 1884.
1749064495: mosquitto version 2.0.21 running
1749064507: New connection from ::1:52852 on port 1884.
1749064507: New client connected from ::1:52852 as auto-B394C14E-CB5E-3065-72D9-FEE164BB34EE (p2, c1, k60, u'aluno').
1749064507: No will message specified.
1749064507: Sending CONNACK to auto-B394C14E-CB5E-3065-72D9-FEE164BB34EE (0, 0)
1749064507: Received SUBSCRIBE from auto-B394C14E-CB5E-3065-72D9-FEE164BB34EE
1749064507: escola/sala1/temperatura (QoS 0)
1749064507: auto-B394C14E-CB5E-3065-72D9-FEE164BB34EE 0 escola/sala1/temperatura
1749064507: Sending SUBACK to auto-B394C14E-CB5E-3065-72D9-FEE164BB34EE
1749064521: New connection from 192.168.137.249:60444 on port 1884.
1749064521: New client connected from 192.168.137.249:60444 as bitdog1 (p2, c1, k0, u'aluno').
1749064521: No will message specified.
1749064521: Sending CONNACK to bitdog1 (0, 0)
1749064521: Received PUBLISH from bitdog1 (d0, q0, r0, m0, 'escola/sala1/temperatura', ... (4 bytes))
1749064521: Sending PUBLISH to auto-B394C14E-CB5E-3065-72D9-FEE164BB34EE (d0, q0, r0, m0, 'escola/sala1/temperatura', ..
. (4 bytes))
```

O recebimento das mensagens pelos clientes autorizados também foi confirmado.

```
C:\Users\gasor>mosquitto_sub -h localhost -p 1884 -t "escola/sala1/temperatura" -u "aluno" -P "senha123"
26.5
26.5
```

Além disso, foram realizados testes de publicação a partir do computador, nos quais foi possível observar que tentativas de envio com senha incorreta foram devidamente recusadas pelo broker.

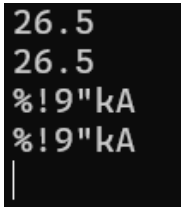
```
C:\Users\gasor>mosquitto_pub -h localhost -p 1884 -t "escola/sala1/temperatura" -u "aluno" -P "senha12" -m "35.6"
Connection error: Connection Refused: not authorised.
Error: The connection was refused.

C:\Users\gasor>mosquitto_pub -h localhost -p 1884 -t "escola/sala1/temperatura" -u "aluno" -P "senha123" -m "35.6"
```

Etapa 5: Simulando criptografia leve (XOR)

Nesta etapa, foi empregada uma técnica simples de criptografia para ofuscar os dados, de modo que apenas quem possui a chave de decodificação pudesse acessar o valor original. Para isso, utilizou-se a operação XOR com o número 23 como chave. Dessa forma, o valor recebido difere da mensagem original, como pode ser observado na imagem a seguir.

Neste caso, o valor recebido será diferente da mensagem original, como pode ser observado na imagem a seguir.



```
26.5
26.5
%!9"kA
%!9"kA
|
```

Observação: a exibição do console está em ASCII e, ao utilizar o valor original (42), o resultado era apenas uma quebra de linha. Por esse motivo, a chave foi alterada para proporcionar uma representação mais clara do processo(23).

De posse do dado recebido e conhecendo o método e a chave de criptografia utilizados, é possível que o destinatário recupere o valor real enviado, aplicando novamente a operação XOR com a mesma chave.

Etapa 6: Proteção contra replay

Com a implementação de mecanismos de proteção contra ataques de replay, caso um dado recebido seja idêntico a outro já processado anteriormente, ele será identificado como replay e, portanto, não será considerado uma nova leitura válida pelo sistema. A imagem a seguir ilustra o resultado dessa funcionalidade.

```
Mensagem recebida no tópico: escola/sala1/temperatura (tam: 21)
Nova leitura: 26.50 (ts: 2)
Mensagem recebida no tópico: escola/sala1/temperatura (tam: 21)
Replay detectado (ts: 2 <= 2)
```

Conclusão

Entre as técnicas apresentadas para comunicação e segurança em IoT, destacam-se como mais escaláveis aquelas que adotam padrões abertos, arquitetura modular e boas práticas de gestão de dispositivos. Protocolos como MQTT, amplamente suportados por plataformas como a BitDogLab, são reconhecidos por sua leveza, eficiência e fácil integração em ambientes com dezenas ou centenas de dispositivos, como ocorre em redes escolares.

A autenticação básica de clientes, controle de acesso por senha e o uso de tópicos segmentados são medidas iniciais que funcionam bem em ambientes educacionais de pequeno a médio porte. Para escalar com segurança, recomenda-se evoluir para autenticação baseada em certificados digitais, uso de criptografia ponta a ponta e políticas centralizadas de controle de acesso (RBAC), que podem ser gerenciadas por servidores dedicados ou soluções de gerenciamento de dispositivos (MDM/EDR). Essas abordagens evitam gargalos de performance e reduzem o risco de falhas de segurança à medida que o número de dispositivos cresce.

No contexto de uma rede escolar com múltiplas BitDogLab, a aplicação dessas técnicas deve considerar:

Segmentação de rede: Separar os dispositivos IoT em VLANs específicas, isolando-os do tráfego geral da escola e facilitando o monitoramento e controle de acesso.

Gerenciamento centralizado: Utilizar plataformas de gerenciamento para registrar, monitorar e atualizar todas as BitDogLab, garantindo que apenas dispositivos autorizados estejam ativos e recebam atualizações de firmware e políticas de segurança.

Padronização de mensagens e protocolos: Adotar formatos de dados e protocolos padronizados (como MQTT sobre TLS) para garantir interoperabilidade e facilitar a integração com sistemas de análise e dashboards educacionais.

Monitoramento contínuo: Implementar ferramentas de monitoramento para detectar anomalias, tentativas de ataque ou falhas de comunicação em tempo real, permitindo resposta rápida e manutenção preventiva.

Automação de tarefas de segurança: Automatizar processos como distribuição de senhas, rotação de chaves e aplicação de atualizações, reduzindo o esforço manual e aumentando a confiabilidade da infraestrutura.

A BitDogLab, por sua arquitetura aberta e flexível, facilita a adoção dessas práticas, permitindo que escolas implementem projetos de IoT escaláveis, seguros e integrados ao currículo, promovendo o aprendizado prático e multidisciplinar. Ao combinar protocolos robustos, práticas de segurança e gestão centralizada, é possível expandir a rede para dezenas ou centenas de dispositivos sem perder o controle, a eficiência ou a segurança operacional.