Proteção de Identificadores em Beacons BLE contra Escuta Passiva usando ESP32 e Android

Everton Costa, Raimundo Barreto, Marcos Maia, Alex Monteiro

[1] Universidade Federal do Amazonas (UFAM) – Instituto de Computação (IComp)

estevam.costa@ufam.edu.br

{marcos.maia, alex.monteiro, rbarreto}@icomp.ufam.edu.br

Abstract. *Bluetooth Low Energy (BLE) wearables are used for identification, attendance, and access control, but advertising packets may expose static identifiers. This paper presents a low-cost prototype for encrypted BLE beacon transmission using an ESP32 wearable and Android gateway. A 16-byte identifier is encrypted with AES-128 using TinyAES and placed in the Manufacturer Data field. The gateway scans beacons, decrypts the payload, and registers events at an RSSI proximity threshold. In an attendance case study, an ESP32 sniffer emulated passive eavesdropping. Results show that the gateway recovers the identifier, while the sniffer captures only encrypted bytes.*

Resumo. *Dispositivos vestíveis com Bluetooth Low Energy (BLE) são usados para identificação, controle de presença e acesso, mas os pacotes de advertising podem expor identificadores estáticos. Este trabalho apresenta um protótipo de baixo custo para transmissão cifrada de beacons BLE usando um vestível com ESP32 e um gateway Android. Um identificador de 16 bytes é cifrado com AES-128 usando TinyAES e inserido no campo Manufacturer Data. O gateway detecta os beacons, decifrar o payload e registra eventos quando o RSSI indica proximidade. Em um estudo de caso de controle de presença, um ESP32 atuou como sniffer para simular escuta passiva. Os resultados mostram que o gateway recupera o identificador, enquanto o sniffer captura apenas bytes cifrados.*

1. Introdução

Dispositivos vestíveis com Bluetooth Low Energy (BLE) são amplamente usados em aplicações de identificação, presença e controle de acesso, especialmente por meio de beacons, que transmitem pacotes de *advertising* sem exigir conexão persistente [Bluetooth SIG 2016, Tanenbaum et al. 2021, Padgette et al. 2017]. Essa característica simplifica a implantação e reduz o custo, mas também cria um problema de segurança: pacotes de *advertising* podem ser capturados por receptores BLE próximos. Quando o identificador do usuário trafega em texto claro, o sistema fica exposto a escuta passiva, rastreamento, clonagem simples e ataques de repetição.

Este artigo apresenta uma arquitetura de baixo custo para transmissão criptografada de beacons BLE usando ESP32 e smartphone Android. A pulseira ESP32 cifra um identificador de usuário com AES-128, usando TinyAES, e insere o payload cifrado no

campo *Manufacturer Data* [Kokke , Bluetooth SIG]. O gateway Android realiza varredura BLE, extrai e decifra o payload, e registra a presença apenas quando o RSSI indica proximidade suficiente.

A proposta é tratada como prova de conceito funcional, não como protocolo criptográfico completo. O objetivo principal é reduzir a exposição direta do identificador contra escuta passiva simples, mantendo baixo custo e facilidade de reprodução em ambiente acadêmico. Ataques ativos, como replay, clonagem e manipulação de payload, são discutidos como riscos remanescentes que exigem mecanismos adicionais de autenticação e gerenciamento de chaves.

Diferentemente de soluções completas de identidade temporária, a versão atual não implementa resolução remota de identificadores, rotação segura de chaves nem proteção contra replay. Essa delimitação torna o protótipo adequado para experimentação acadêmica e iniciação científica, mas também exige cuidado para não apresentar a solução como proteção criptográfica completa.

As principais contribuições deste trabalho são: (1) uma arquitetura de baixo custo para envio de identificadores cifrados em beacons AES em BLE advertising usando ESP32 e gateway Android; (2) uma implementação funcional com três componentes: pulseira transmissora, gateway legítimo e sniffer BLE para avaliação ofensiva simples; (3) um estudo de caso educacional, com identificadores sintéticos, mostrando a diferença entre capturar o pacote BLE e recuperar o identificador em texto claro; e (4) uma análise explícita das limitações criptográficas da versão atual, incluindo IV, autenticação, replay, clonagem e gerenciamento de chaves.

2. Fundamentação Teórica

BLE é uma tecnologia de comunicação sem fio projetada para baixo consumo energético, sendo amplamente utilizada em dispositivos vestíveis, sensores e aplicações de Internet das Coisas (IoT). Em diversos cenários, dispositivos BLE operam por meio de pacotes de *advertising*, transmitidos periodicamente para anunciar presença, disponibilizar informações ou permitir descoberta por dispositivos próximos. Esse modelo favorece aplicações como identificação, rastreamento, monitoramento de presença e controle de acesso, pois dispensa pareamento ou conexão persistente entre os dispositivos [Bluetooth SIG 2016, Padgette et al. 2017].

O campo *Manufacturer Data* permite transportar dados específicos da aplicação dentro do pacote de *advertising*. Em protótipos, esse campo costuma ser usado para incluir um identificador de dispositivo, um código de usuário ou uma credencial lógica [Bluetooth SIG]. Contudo, o canal de *advertising* não deve ser tratado como privado. Qualquer dispositivo BLE próximo pode observar os pacotes transmitidos. Portanto, identificadores fixos em texto claro podem permitir rastreamento, associação indevida de identidade e falsificação de presença.

Para reduzir esse risco, este trabalho adota criptografia simétrica com AES-128. A biblioteca TinyAES foi escolhida por ser compacta, portátil e adequada a dispositivos embarcados com restrições de memória e processamento [Kokke]. O uso de AES protege a confidencialidade do payload transmitido, mas não resolve sozinho todos os requisitos de segurança. Uma solução robusta também precisa tratar geração correta de vetor de

inicialização (IV), autenticação da mensagem, gerenciamento de chaves e proteção contra replay [Dworkin 2001, Dworkin 2005, Dworkin 2007, Stallings 2014].

3. Trabalhos Relacionados

A segurança em Bluetooth envolve mecanismos de pareamento, autenticação, confidencialidade e privacidade, mas nem todos esses mecanismos se aplicam diretamente a beacons baseados apenas em *advertising*. O guia NIST SP 800-121 Rev. 2 destaca que Bluetooth é uma tecnologia de rádio de curto alcance amplamente usada em dispositivos pessoais e corporativos, e recomenda que sistemas baseados nessa tecnologia considerem tanto os recursos de segurança do padrão quanto os riscos do ambiente de implantação [Padgette et al. 2017].

Protocolos de beacon também tratam a exposição de identificadores como uma preocupação de privacidade. A especificação Eddystone define diferentes tipos de quadros para beacons BLE, incluindo quadros voltados à transmissão de identificadores e telemetria [Google]. Uma evolução relevante é o Eddystone-EID, que introduz identificadores efêmeros para reduzir rastreamento por identificadores estáticos. Nesse modelo, o identificador muda periodicamente e só é útil para entidades autorizadas a resolvê-lo [Google Security Blog 2016]. A proposta deste artigo é mais simples e tem outro foco: construir uma prova de conceito de baixo custo, baseada em ESP32 e Android, que cifra o payload do beacon e permite avaliação com um *sniffer* BLE.

4. Modelo de Ameaça e Requisitos

Considera-se um cenário em que uma pulseira BLE transmite beacons periódicos para um gateway Android, responsável por reconhecer dispositivos autorizados e registrar eventos de presença ou acesso. A versão atual foca em um atacante passivo local, capaz de capturar pacotes de *advertising* com dispositivos BLE, como ESP32, smartphone ou notebook, buscando obter o identificador transmitido.

Ataques ativos, como replay, clonagem, manipulação de payload e roubo de chave, são considerados fora do escopo implementado nesta prova de conceito, mas são discutidos como limitações importantes. A mitigação dessas ameaças exige mecanismos adicionais, como contador, nonce, timestamp seguro, código de autenticação de mensagem (MAC), CMAC, HMAC ou modos de cifra autenticada.

Em relação ao baixo consumo energético, cabe ressaltar que a escolha do ESP32 impõe desafios diretos ao projeto físico do dispositivo vestível. Como o microcontrolador apresenta picos de consumo elétrico ao acionar o rádio BLE, o gerenciamento de energia vai além da escolha do protocolo, exigindo a adoção de modos de economia de energia (como o Deep Sleep) nos intervalos de transmissão. Essa otimização entre hardware e firmware é crucial para viabilizar a operação prolongada da pulseira alimentada por baterias portáteis de pequena capacidade.

5. Arquitetura Proposta

A arquitetura proposta foi projetada considerando as limitações típicas de dispositivos embarcados de baixo custo, como o ESP32. Nesse contexto, decisões como o uso de BLE advertising, payload compacto no campo Manufacturer Data, criptografia simétrica AES-128 e biblioteca TinyAES refletem um compromisso entre segurança, simplicidade de

Tabela 1. Requisitos de segurança e operação do protótipo.

Requisito	Descrição
R1 – Confidencialidade do identificador	O identificador do usuário não deve trafegar em texto claro nos pacotes BLE.
R2 – Baixo custo	A solução deve usar hardware acessível, como ESP32 e smartphone Android.
R3 – Operação sem pareamento	O gateway deve detectar a pulseira sem exigir pareamento BLE manual.
R4 – Filtragem de dispositivos	O gateway deve processar apenas beacons compatíveis com o padrão esperado.
R5 – Registro por proximidade	O evento deve ser registrado apenas quando o RSSI indicar proximidade suficiente.
R6 – Avaliação ofensiva	O sistema deve ser testado contra escuta passiva usando um sniffer BLE.
R7 – Evolução para proteção ativa	A arquitetura deve permitir inclusão futura de proteção contra replay, clonagem e manipulação de payload.

implementação, consumo energético e viabilidade prática. Assim, o objetivo do protótipo não é substituir soluções completas de autenticação usadas em ambientes críticos, mas demonstrar uma alternativa leve e reprodutível para reduzir a exposição de identificadores em aplicações de controle de presença e acesso baseadas em beacons BLE. A Figura 1 apresenta a visão geral da arquitetura. A solução possui três componentes: pulseira transmissora, gateway Android e sniffer de avaliação.

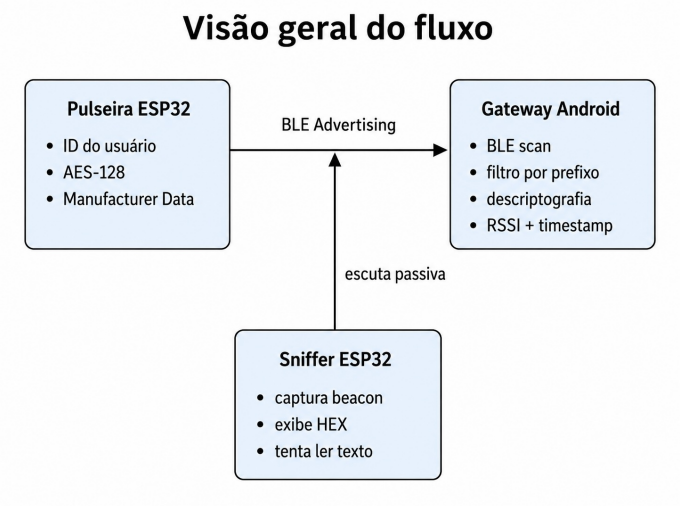

Figura 1. Visão geral da arquitetura proposta.

5.1. Pulseira Transmissora

A pulseira transmissora é baseada no microcontrolador ESP32. Sua função é gerar periodicamente um beacon BLE contendo um identificador criptografado. O identificador representa uma credencial lógica do usuário, como um código de aluno, funcionário ou

participante. Para fins de protótipo, foi usado um identificador textual sintético de 16 bytes, ajustado ao tamanho de um bloco AES. O fluxo da pulseira é: (i) inicializar a pilha BLE; (ii) definir o nome do dispositivo reconhecível pelo gateway; (iii) preparar o identificador do usuário; (iv) cifrar o identificador usando AES-128; (v) inserir o payload cifrado no campo *Manufacturer Data*; e (vi) iniciar a transmissão periódica dos beacons.

5.2. Gateway Android

O gateway Android recebe e processa os beacons. A escolha de um smartphone reduz o custo da solução, pois dispensa hardware adicional na fase inicial do projeto. Além disso, smartphones Android já possuem recursos de BLE scanning, processamento criptográfico, armazenamento local e interface com usuário. O fluxo do gateway é: (i) realizar varredura BLE contínua ou periódica; (ii) filtrar dispositivos cujo nome segue o prefixo esperado; (iii) extrair o campo *Manufacturer Data*; (iv) remover bytes auxiliares de identificação do fabricante; (v) aplicar a decifragem com a chave pré-compartilhada; (vi) recuperar o identificador em texto claro; (vii) medir o RSSI do pacote recebido; e (viii) registrar o evento quando o RSSI atinge o limiar de proximidade.

5.3. Sniffer BLE

O sniffer BLE foi implementado em outro ESP32 para simular um atacante passivo. Ele realiza varredura BLE, filtra dispositivos com o prefixo usado pela pulseira e extrai o *Manufacturer Data*. O sniffer tenta exibir o payload como texto e como bytes em hexadecimal. Esse componente permite observar se o identificador aparece em texto claro ou se apenas o conteúdo cifrado é capturado.

6. Implementação

6.1. Formato do Payload

O protótipo utiliza um identificador textual sintético de 16 bytes, compatível com o tamanho de bloco do AES. Esse identificador é cifrado antes de ser inserido no pacote BLE. O campo *Manufacturer Data* possui 18 bytes no protótipo: os 2 primeiros bytes são usados como identificador de fabricante em ambiente experimental, e os 16 bytes restantes correspondem ao identificador cifrado com AES-128. Em uma versão de produção, o identificador de fabricante deve seguir as atribuições formais da especificação Bluetooth, evitando o uso de valores meramente experimentais [Bluetooth SIG]. A estrutura atual foi mantida por simplicidade, pois o foco foi validar o fluxo de transmissão, interceptação e recuperação legítima do identificador.

6.2. Cifragem no ESP32

A pulseira utiliza AES-128 para cifrar o identificador. O procedimento geral é apresentado na Figura 2. A versão inicial usa uma chave pré-compartilhada definida no firmware e no aplicativo gateway. Essa decisão facilita a validação funcional, mas não é suficiente para implantação real em larga escala. Uma versão posterior deve incluir provisionamento seguro de chaves, rotação periódica, revogação e armazenamento protegido no smartphone. Quando AES-CBC é usado, o vetor de inicialização (IV - Initialization Vector) não deve ser tratado como constante fixa para todas as mensagens. A recomendação NIST SP 800-38A indica que, nos modos CBC e CFB, o vetor de inicialização deve ser imprevisível para cada execução da cifragem; ele não precisa ser secreto e pode ser transmitido

```
Entrada:
  ID_usuario: identificador sintetico de 16 bytes
  K: chave simetrica de 128 bits
  IV: vetor de inicializacao

Procedimento:
  buffer <- ID_usuario
  contexto <- AES_init(K, IV)
  C <- AES_CBC_encrypt(buffer)
  manufacturerData <- companyId || C
  transmitir manufacturerData via BLE advertising
```

Figura 2. Procedimento simplificado de cifragem e transmissão.

junto com o texto cifrado, desde que seja gerado corretamente [Dworkin 2001]. Portanto, a versão atual deve ser entendida como prova de conceito. Para evoluir a segurança, recomenda-se incluir um vetor de inicialização variável, contador ou nonce, ou migrar para modo autenticado, como AES-GCM, quando as restrições de pacote e biblioteca permitirem [Dworkin 2007].

6.3. Recepção e Decifragem no Gateway

O aplicativo Android realiza varredura BLE e processa apenas dispositivos cujo nome começa com um prefixo pré-definido. Depois de identificar um beacon compatível, extrai o *Manufacturer Data*. Os dois primeiros bytes são tratados como cabeçalho experimental, e os 16 bytes seguintes são tratados como payload criptografado. Em seguida, o aplicativo aplica a decifragem com a chave simétrica compartilhada. Além da decifragem, o gateway usa RSSI como critério simples de proximidade. No protótipo, o registro oficial do evento ocorre quando o sinal atinge aproximadamente -30 dBm. Leituras abaixo desse limiar indicam que a pulseira foi detectada, mas ainda não foi considerada suficientemente próxima para registrar presença ou acesso.

```
[ALERTA] Dispositivo alvo encontrado
[-] Interceptando pacote BLE advertising
[-] Tentativa de leitura em texto: caracteres nao legiveis
[-] Payload interceptado em HEX: A3 7F 21 9C ...
```

Figura 3. Saída simplificada do sniffer BLE usado na avaliação funcional.

6.4. Sniffer de Avaliação

O sniffer aplica o mesmo filtro por prefixo de nome e extrai o *Manufacturer Data* dos pacotes encontrados. Como mostrado na Figura 3, ele tenta interpretar o payload como texto e também o exibe em hexadecimal. Esse comportamento demonstra a diferença entre capturar o pacote e compreender seu conteúdo. A captura é possível porque BLE advertising é observável por receptores próximos; a recuperação do identificador, porém, exige a chave correta.

7. Estudo de Caso e Avaliação Funcional

Foi realizado um estudo de caso funcional em um cenário educacional de controle de presença. A pulseira ESP32 transmitiu beacons BLE criptografados para um smartphone Android, que atuou como gateway receptor. O estudo verificou a transmissão de um identificador cifrado, sua correta decifragem, o registro de presença apenas em proximidade suficiente e a proteção contra leitura em texto claro por sniffers BLE. Foram usados identificadores sintéticos, sem dados reais de estudantes, reduzindo riscos de privacidade. Em uma implantação real, seriam necessários cuidados adicionais com base legal, minimização de dados, controle de acesso e políticas de retenção.

7.1. Procedimento Experimental

O experimento verificou o funcionamento da arquitetura e seu comportamento em diferentes condições de proximidade. A pulseira ESP32 transmitiu periodicamente o identificador sintético cifrado via BLE, enquanto o gateway Android realizou a varredura e processou apenas dispositivos com o prefixo definido. Um segundo ESP32 atuou como sniffer BLE para capturar os pacotes e analisar o payload interceptado em texto e hexadecimal.

Foram avaliados quatro cenários de distância entre a pulseira e o gateway: 0,5 m, 1 m, 3 m e 6 m. Em cada um, realizaram-se 50 transmissões, registrando pacotes detectados, payloads decifrados corretamente, RSSI médio e ocorrência de presença. O protótipo adotou limiar de proximidade de cerca de -30 dBm, considerando como presença válida apenas leituras próximas a esse valor.

7.2. Resultados Funcionais Observados

A Tabela 2 resume os resultados funcionais observados. Os resultados mostram que o fluxo básico foi validado: a pulseira transmite um identificador protegido, o gateway recupera o identificador e o sniffer captura o pacote sem exibir o identificador em texto claro.

7.3. Resultados Quantitativos Preliminares

A Tabela 3 apresenta uma avaliação quantitativa simples, com valores preliminares usados como referência experimental. Em cada distância, foram consideradas 50 transmissões de beacon. A coluna *Detectados* indica quantos pacotes foram observados pelo gateway Android. A coluna *Decifrados* indica quantos pacotes detectados tiveram o identificador sintético recuperado corretamente. A coluna *RSSI médio* resume a intensidade média do sinal observada no gateway. A coluna *Registro correto* indica se o comportamento do sistema foi compatível com a lógica esperada de proximidade.

8. Discussão

O estudo de caso mostra que a arquitetura é viável como prova de conceito de baixo custo. A integração entre ESP32, BLE, AES-128, Android e sniffer BLE funcionou no fluxo básico proposto. Esse resultado é relevante para cenários acadêmicos e de iniciação científica, pois permite estudar segurança em IoT com componentes acessíveis e reproduzíveis.

Tabela 2. Resultados funcionais do estudo de caso.

Questão	Resultado observado	Interpretação
QA1	A pulseira transmitiu beacons BLE contendo dados no campo *Manufacturer Data*.	O firmware do ESP32 operou como beacon criptográfico.
QA2	O gateway detectou o beacon, extraiu o payload e recuperou o identificador após a decifragem.	O fluxo legítimo de transmissão, recepção e decifragem foi validado.
QA3	O aplicativo diferenciou detecção de registro usando RSSI como critério de proximidade.	O sistema representou uma lógica prática de presença baseada em aproximação.
QA4	O sniffer BLE capturou pacotes transmitidos pela pulseira.	O teste confirmou que BLE advertising pode ser observado por dispositivos próximos.
QA5	O sniffer exibiu apenas bytes cifrados, sem o identificador em texto claro.	A criptografia reduziu a exposição contra escuta passiva simples.

Tabela 3. Resultados quantitativos preliminares por distância.

Exp.	Distância	Enviados	Detectados	Decifrados	RSSI médio / registro
E1	0,5 m	50	50 (100%)	50 (100%)	-54 dBm / sim
E2	1 m	50	50 (100%)	50 (100%)	-60 dBm / sim
E3	3 m	50	45 (90%)	45 (100%)	-70 dBm / não
E4	6 m	50	39 (78%)	39 (100%)	-74 dBm / não

Os resultados quantitativos preliminares da Tabela 3 reforçam essa interpretação. Nas menores distâncias, o gateway apresentou maior taxa de detecção dos beacons, chegando a 100% em 0,5 m e 1 m. Em distâncias maiores, a taxa de detecção diminuiu, ficando em 90% a 3 m e 78% a 6 m. Esse comportamento é coerente com o uso de BLE advertising em ambientes reais, nos quais a recepção tende a variar conforme distância, obstáculos, orientação da pulseira e interferências.

A taxa de decifragem foi de 100% entre os pacotes detectados nos quatro cenários considerados. Esse resultado sugere que, uma vez recebido corretamente o payload criptografado, o gateway consegue recuperar o identificador sintético de forma consistente. Portanto, as perdas observadas na avaliação preliminar estão mais relacionadas à detecção do beacon e à variação do sinal do que à rotina de decifragem em si.

Do ponto de vista de segurança, o principal resultado continua sendo a redução da exposição direta do identificador. O atacante passivo consegue capturar o pacote BLE, mas não recupera o identificador em texto claro sem a chave secreta. Assim, a proposta diferencia captura de compreensão do conteúdo transmitido. Esse ganho, porém, deve ser interpretado com cuidado: a cifragem contribui para confidencialidade, mas não garante integridade, autenticidade, nem resistência a retransmissão.

O uso de RSSI como critério de proximidade também deve ser visto como uma solução prática, não como medida precisa de distância. Na avaliação preliminar, apenas os cenários de 0,5 m e 1m atingiram o limiar aproximado de -60 dBm e resultou em registro de presença. Nos demais cenários, a pulseira foi detectada, mas o evento não foi registrado. Esse comportamento é desejável para evitar registros indevidos de dispositivos apenas presentes no ambiente, mas mostra que o limiar precisa ser calibrado para cada local. Um limiar muito rígido pode causar falsos negativos, enquanto um limiar muito permissivo pode gerar registros indevidos em salas próximas ou corredores.

8.1. Limitações Criptográficas e Riscos remanescentes

A versão atual usa chave pré-compartilhada fixa entre pulseira e gateway. Essa escolha simplifica a validação inicial, mas não escala bem para muitos usuários. Se a mesma chave for usada em várias pulseiras, o vazamento de uma única chave pode comprometer todo o sistema. Uma versão mais robusta deve usar chaves próprias por pulseira ou chaves derivadas de modo seguro.

Outra limitação está no uso de AES-CBC. Se o vetor de inicialização (IV) for fixo ou reutilizado, mensagens iguais podem gerar padrões repetidos, e o texto cifrado pode se tornar um identificador estático observável. Além disso, cifragem sem autenticação não impede manipulação, replay ou clonagem. Para reduzir esses riscos, versões futuras devem incluir autenticação de mensagem, como CMAC [Dworkin 2005], ou modo autenticado, como AES-GCM [Dworkin 2007], respeitando as restrições de tamanho dos pacotes BLE.

No ataque de replay, um pacote BLE válido pode ser capturado e retransmitido posteriormente, levando o gateway a registrar presença ou acesso indevido. A clonagem também é possível quando outro dispositivo usa o mesmo prefixo de nome e retransmite dados capturados. Portanto, a filtragem por prefixo deve ser tratada apenas como seleção de candidatos, e não como autenticação do dispositivo. Mecanismos como contador monotônico, nonce, timestamp ou desafio-resposta, são necessários para avançar além da proteção contra escuta passiva.

8.2. Ameaças à Validade

Há quatro ameaças à validade principais.

Validade interna: Os testes foram realizados em ambiente controlado, com poucos dispositivos e cenários reduzidos. Interferências de outros dispositivos BLE, obstáculos físicos e variações entre smartphones podem alterar o comportamento observado.

Validade externa: Os resultados não devem ser generalizados diretamente para ambientes reais de grande escala. Salas com muitos alunos, empresas com várias portas ou eventos com centenas de participantes podem apresentar colisões, interferência e atrasos não observados nesta etapa.

Validade de construção: A avaliação mede principalmente funcionamento do fluxo e confidencialidade contra escuta passiva simples. Ainda não mede autenticação forte, resistência a replay, clonagem, integridade do payload e robustez contra atacante ativo.

Validade conclusiva: Embora o artigo inclua uma tabela quantitativa preliminar, a análise ainda precisa ser expandida com repetições controladas, diferentes ambientes, múltiplos

dispositivos e análise estatística. O trabalho demonstra viabilidade técnica, mas ainda exige experimentos adicionais para sustentar afirmações mais fortes sobre desempenho e segurança operacional.

9. Conclusão

Este artigo apresentou uma arquitetura de baixo custo para transmissão criptografada de beacons BLE usando ESP32 e gateway Android. O protótipo desenvolvido demonstra que é possível cifrar um identificador sintético no dispositivo vestível, transmiti-lo no campo *Manufacturer Data* de um pacote BLE e recuperá-lo em um aplicativo Android autorizado.

Um aspecto importante da solução é que suas escolhas técnicas foram condicionadas pelas restrições do hardware utilizado. O ESP32 oferece suporte a BLE e capacidade suficiente para executar criptografia simétrica, mas ainda opera em um contexto de recursos limitados quando comparado a servidores, smartphones ou gateways dedicados. Além disso, o uso de BLE advertising impõe restrições de tamanho ao payload transmitido, o que influencia diretamente a escolha dos campos enviados, do modo de cifragem e dos mecanismos adicionais de segurança. Por isso, a solução proposta deve ser entendida como uma abordagem leve, voltada a dispositivos embarcados de baixo custo, e não como uma arquitetura final para autenticação forte em ambientes de alta criticidade.

A avaliação funcional indica que um atacante passivo consegue capturar o beacon, mas não consegue ler diretamente o identificador em texto claro sem a chave secreta. Também foi demonstrado que o gateway detecta o dispositivo, decifra o payload e registra eventos com base em um limiar de RSSI. O principal resultado é a validação inicial de uma arquitetura simples, acessível e reprodutível para experimentação em segurança de beacons BLE.

Como trabalhos futuros, pretende-se expadir o experimento com situações reais, coletando mais dados sobre taxa de detecção, variação de RSSI, taxa de decifragem, tempo de resposta, consumo energético e autonomia da pulseira. Também é necessário fortalecer o protocolo com vetor de inicialização variável, autenticação de mensagem e gerenciamento adequado de chaves, incluindo criação, armazenamento, distribuição, renovação e revogação.

Agradecimentos

Os autores agradecem à Pró-Reitoria de Inovação Tecnológica da Universidade Federal do Amazonas (UFAM), responsável pela coordenação do Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação (PIBITI). Agradecem também à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES/PROEX) – Código de Financiamento 001. Este trabalho foi parcialmente financiado pela Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM), por meio do projeto POSGRAD 2026/2027.

Referências

Bluetooth SIG. Supplement to the Bluetooth Core Specification: Data Types Specification.

Bluetooth SIG (2016). Bluetooth Core Specification Version 5.0.

Dworkin, M. (2001). Recommendation for Block Cipher Modes of Operation: Methods and Techniques. NIST Special Publication 800-38A, National Institute of Standards and Technology, Gaithersburg, MD, USA.

Dworkin, M. (2005). Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38B, National Institute of Standards and Technology, Gaithersburg, MD, USA.

Dworkin, M. (2007). Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D, National Institute of Standards and Technology, Gaithersburg, MD, USA.

Google. Eddystone: An Open Beacon Format. Repositório de software.

Google Security Blog (2016). Growing Eddystone with Ephemeral Identifiers: A Privacy Aware and Secure Open Beacon Format. Blog técnico.

Kokke, K. Tiny AES in C. Repositório de software.

Padgette, J., Scarfone, K., and Chen, L. (2017). Guide to Bluetooth Security. NIST Special Publication 800-121 Revision 2, National Institute of Standards and Technology, Gaithersburg, MD, USA.

Stallings, W. (2014). *Criptografia e Segurança de Redes: Princípios e Práticas*. Pearson Universidades, 6 edition.

Tanenbaum, A. S., Feamster, N., and Wetherall, D. J. (2021). *Redes de Computadores*. Bookman, 6 edition.