Avaliação de Protocolos de Acordo de Chave Baseados em Sinais Fisiológicos para Redes Corporais sem Fio

Caio V. F. Silva, Samuel P. C. Sena, Kristtopher K. Coelho, Edelberto F. Silva,
Alex B. Vieira, Michele Nogueira, José Augusto M. Nacif

***Resumo*—Os dispositivos em uma rede corporal sem fio monitoram sinais clínicos de pacientes. A importância contida nestes dados sugerem que devam ser transmitidos e mantidos de forma privada e segura. Com base nos princípios de segurança e as especificidades da rede, este trabalho objetiva apresentar uma análise empírica acerca do desempenho e consumo de recursos de hardware por protocolos de autenticação. Foram realizadas análises acerca de tempo de execução, consumo de memória, *goodput*, além das métricas de falsa aceitação e falsa rejeição sobre tentativas de autenticação. Ao explorar os resultados, é notável o *trade-off* entre os protocolos, principalmente sobre tempo de execução e consumo de memória, além da precisão durante o processo de autenticação.**

***Index Terms*—Protocolos de Acordo de Chave, Sinais Biométricos, ECG, PPG, Redes Corporais sem Fio**

I. Introdução

A rede mundial de computadores cresce dia a dia, e parte desse crescimento constante deve-se à evolução das Redes de Sensores Sem Fio (*Wireless Sensor Network* - WSN). As WSNs são utilizadas em diversos domínios, tais como agricultura, urbanização, segurança e medicina [1]. No cenário clínico, utiliza-se uma WSN para realizar monitoramento proativo de pacientes, permitindo o tratamento a distância de modo contínuo. Atuando de forma cooperativa entre si, dispositivos implantáveis ou vestíveis de monitoramento de sinais vitais compõem um tipo de WSN denominada Rede Corporal Sem Fio (*Wireless Body Area Network* - WBAN) [2].

Os dispositivos em uma rede WBAN monitoram os sinais fisiológicos dos pacientes, tais como: pressão arterial, temperatura, eletrocardiograma (ECG), sinais de fotopletismografia (PPG), entre outros. Estes dados aferidos são encaminhados para centros avançados de monitoramento com profissionais especializados em cuidados de saúde [3]. A importância contida nos dados sugerem que devam ser transmitidos e mantidos de forma privada e segura, de modo a inibir ameaças externas. Os dados aferidos são informações individuais e sensíveis, onde quaisquer alterações podem afetar o diagnóstico de doenças e tratamento. Portanto, durante o transporte e manipulação, é necessário a utilização de medidas de segurança para garantir a integridade e privacidade destes dados [4].

Com base nos requisitos de segurança e privacidade, protocolos de autenticação de usuários em dispositivos WBAN são indispensáveis. Entretanto, os dispositivos que compõem a WBAN possuem limitações, tais como: baixa capacidade de armazenamento energético, capacidade de transmissão dos dados reduzida, pouco poder de processamento, entre outras [5]. Ainda, uma WBAN deve atuar de forma escalável, resiliente e energeticamente eficiente. Estes requisitos são importantes, principalmente quando conceitua-se aplicações com dispositivos implantáveis. Uma vez que estes dispositivos devam manter-se em pleno funcionamento o maior tempo possível para que não ocorra desconforto recorrente à remoção, substituição ou recarga de um dispositivo inoperante. De maneira complementar, é desejável que uma WBAN tenha a facilidade para adicionar e configurar (*Plug and Play*) novos dispositivos em uma rede [6]. Desta forma, é necessário propor e avaliar protocolos para redes WBAN que atendam os requisitos de segurança citados, e também utilizem de maneira eficiente os recursos de *hardware* disponíveis.

Na literatura é possível encontrar propostas de diversos protocolos para acordos de chaves para WSN [7]. Entretanto, é necessária uma análise mais aprofundada destas soluções, considerando as especificidades de uma WSN corporal, ou WBAN. Sabemos que protocolos que empregam sinais fisiológicos para definirem chaves secretas e compartilhá-las entre dispositivos tendem a atender aos requisitos desejáveis em uma WBAN [3]. Os sinais fisiológicos comumente utilizados em protocolos de estabelecimento de chaves envolvem o ECG e PPG [8]. Também é possível encontrar na literatura a indicação de que métodos de acordo de chaves que seguem primitivas difusa (*Fuzzy Primitive*), ou a primitivas não difusa (*Non-fuzzy Primitive*) podem ser interessantes [9]. A *Fuzzy Primitive* tem como objetivo esconder um segredo mesclando-o com outros dados. Para isto podem ser utilizadas técnicas de compromisso difuso (*Fuzzy Commitment*) ou cofre difuso (*Fuzzy Vault*). Em métodos conhecidos como *Fuzzy Commitment*, é estabelecida uma chave aleatória, a qual é combinada aos sinais fisiológico

Este trabalho foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq/Brasil), pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brasil), portaria 88887.509309/2020-00 e pela Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG). Caio Silva, Samuel Sena, Kristtopher Coelho e José Augusto M. Nacif são do Instituto de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, Brasil. E-mail: {caio.fernandes, samuel.sena, kristtopher.coelho, jnacif}@ufv.br. Alex Vieira e Edelberto Silva são do Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora, Brasil. E-mail: alex.borges@ufjf.edu.br e edelberto@ice.ufjf.br. Michele Nogueira é do Departamento de Ciência da Computação da Universidade Federal de Minas Gerais, Brasil. E-mail: michele@dcc.ufmg.br.

do usuário [10]. Para a classe de algoritmos *Fuzzy Vault*, a autenticação é dada pela reconstrução de um polinômio. Tal polinômio é inicialmente elaborado com características de sinais fisiológicos, adicionado de pontos falsos, os quais são produzidos aleatoriamente [11]. A *Non-fuzzy Primitive* engloba estratégias que não utilizam técnicas para combinar um segredo aleatório, porém ainda utilizam sinais fisiológicos para estabelecer o acordo. Como por exemplo pode-se destacar [12], que gera uma chave a partir do próprio sinal fisiológico. Em [13], utiliza-se o modelo oculto de Markov (*Hidden Markov Model*-HMM). Além de [14], que usa a mistura gaussiana (*Gaussian Mixture Model*-GMM).

Assim, a principal contribuição deste trabalho é a apresentação de uma análise empírica de desempenho comparativa entre os dois principais métodos citados em um ambiente WBAN. Um que segue uma *Fuzzy Primitive*, e outro esquema, que segue uma *Non-fuzzy Primitive*. O objetivo é avaliar o custo para ambas as classes de protocolos que empregam sinais fisiológicos para o acordo de chaves utilizando um dispositivo real. Realizamos investigações acerca de tempo de execução, consumo de memória, quantidade de dados úteis transmitidos durante um intervalo (*goodput*), além das métricas de falsa aceitação (*False Acceptance Rate* - FAR) e falsa rejeição (*False Rejection Rate* - FRR) sobre tentativas de autenticação. Ao fim da análise percebe-se que o protocolo *Fuzzy Primitive* apresentou um tempo de execução mais longo, com um *goodput* ligeiramente inferior. Entretanto, consume menos memória. Além disso, constata-se que o protocolo *Non-Fuzzy Primitive* possui uma limitação quanto à sincronização dos dispositivos, uma vez que devam ser perfeitamente sincronizados. Os protocolos implementados para esta avaliação estão disponíveis em repositórios públicos[1][2], de modo a contribuir com a evolução do tema junto a comunidade científica.

A organização do artigo segue como descrita. A Seção II aborda uma discussão sobre trabalhos relacionados. A Seção III, detalha os protocolos de autenticação avaliados. A Seção IV descreve a metodologia da avaliação dos protocolos. Na Seção V os resultados são discutidos e, por fim, a Seção VI exibe as considerações finais e trabalhos futuros.

II. Trabalhos relacionados

Os dispositivos que compõem uma rede corporal possuem diversas limitações, entre elas: baixa capacidade de processamento e armazenamento de dados, além de recursos energéticos escassos e taxa de comunicação reduzida [5]. Portanto, é imprescindível que os protocolos de acordo de chaves atendam a estas necessidades básicas. Além disso, os esquemas de segurança devem atender aos critérios de protocolos biométricos [15], tais como distintividade (distinguível entre pessoas diferentes) e invulnerabilidade (resistente a ataques).

Em [10], é apresentado um esquema de acordo de chaves utilizando biometria para autenticação. Esse esquema segue uma *Fuzzy Primitive* em conjunto com códigos de correção de erro e criptografia para garantir o acordo de chaves em segurança, introduzindo o conceito *Fuzzy Commitment*. Entretanto, a análise de segurança realizada é baseada em dados biométricos como digital de algum dedo e íris. Ademais, o esquema não foi desenvolvido especificamente para redes de dispositivos corporais envolvendo sinais como ECG e PPG. Portanto, surgiram propostas voltadas para WBANs, como [16], que apresenta o protocolo *Physiological-Signal-based Key Agreement* (PSKA), em [12] é demonstrado o protocolo *Electrocardiogram based Key Agreement* (EKA), e o esquema denominado acordo de chave baseado em características fisiológicas ordenadas (*Ordered-Physiological-Feature-based Key Agreement*-OPFKA) proposto por [17].

O protocolo PSKA [16] segue a *Fuzzy Primitive*, utilizando o conceito de *Fuzzy Vault*, introduzido por [11]. O protocolo cumpre os critérios de segurança gerando chaves longas e aleatórias. Utiliza sinal fisiológico distinguível entre pessoas e que varia com o tempo. Além da análise de segurança, realiza também a análise de desempenho, avaliando custo computacional em termos de ciclos de relógio e a quantidade de uso de memória. Sua avaliação é restrita ao próprio protocolo.

Igualmente embasado pelo esquema *Fuzzy Vault*, em [17] é proposto um esquema de acordo de chave que não realiza a reconstrução polinômios ou utiliza códigos de correção de erros. Tal proposta beneficia-se do fato das características produzidas por um dispositivo serem ordenadas e apenas o próprio dispositivo conhecer a ordem. Deste modo, o OPFKA utiliza as características do sinal vital mescladas com ruído para fornecer segurança aprimorada. A avaliação apresentada pelos autores compreende as características da chave secreta produzida, bem como as sobrecargas de armazenamento em memória, comunicação e consumo de energia durante as comunicações. Entretanto, sua comparação é limitada à mesma classe de algoritmos e baseada no protocolo PSKA.

O protocolo EKA [12] proporciona o acordo de chaves aplicando a *Non-Fuzzy Primitive*. Ele foi elaborado para aplicação direta em redes WBAN, e utiliza sinais fisiológicos como ECG para estabelecer o acordo de chaves. Nesta proposta ainda é realizada a análise de segurança do esquema. Perante a análise, são apresentados resultados sobre a distintividade entre pessoas, aleatoriedade e variação temporal da chave, uma vez que os sinais fisiológicos variam com o tempo. Por mais que o trabalho tenha realizado a análise de segurança, uma análise de desempenho não foi realizada. Portanto, existe a necessidade de verificar o comportamento do protocolo em relação a consumo de recursos de hardware.

Considerando os trabalhos listados, nosso trabalho se distingue do estado da arte ao oferecer uma análise empírica comparativa entre as principais etapas de cada protocolo. Esta avaliação considera a utilização de um dispositivo real, evidenciando o consumo de recursos e respectivo desempenho dos protocolos.

III. Protocolos de acordo de chave

Nesta Seção são descritos os protocolos avaliados, apresentando características intrínsecas sobre cada um deles. Como

[1]https://github.com/caiofers/pska2010
[2]https://github.com/caiofers/eka2008

a natureza da análise empírica é uma comparação de desempenho entre protocolos para WBAN de classes distintas, os protocolos selecionados atendem de antemão aos requisitos desejáveis das redes corporais [6]. Dentre os principais requisitos observados lista-se:

- Escalabilidade: Suficientemente escalável de modo que a segurança não seja comprometida ao adicionar um novo nó na rede;
- Resiliência a ataques: Capacidade de se opor à um ataque que deseja capturar um nó da rede;
- Eficiência energética: Baixo consumo ao gerar as chaves e realizar o acordo de chaves entre os dispositivos;
- *Plug-and-play*: Permite a adição automática de novos dispositivos com aptidão para a produzir e realizar acordo de chaves sem interferência de terceiros;

Fundamentado nestes requisitos, os trabalhos eleitos para avaliação foram o *Fuzzy Vault* [16] e [17] descritos nas seções III-A e III-B e o *Electrocardiogram Based (EKG-Based)* (Seção III-C) [12].

A. PSKA

A técnica *Fuzzy Vault* [11] para acordo de chaves consiste em trancar um "segredo" em um cofre. Beneficiando desta técnica, o protocolo PSKA [16] é detalhado em alto nível pela Figura 1. No cofre do PSKA são preservados os pontos $(v, P(v))$, onde v é um valor que pertence ao vetor de características e $P(v)$ é o resultado do polinômio que utiliza v como variável. O polinômio é reconstruído no nó de destino utilizando tais pontos, a fim de concretizar a autenticação.

Os sinais fisiológicos que virão a compor a chave secreta são coletados de forma síncrona durante um intervalo de tempo específico e serão utilizados durante o processo de extração das características. No PSKA os sinais coletados são divididos em janelas. Cada janela é transformada do domínio de tempo para domínio de frequência utilizando a transformada rápida de Fourier (FFT). Posteriormente, tuplas valor-índice são extraídas utilizando um método de detecção de pico. Cada tupla é quantizada para representar um sinal digital. Ao concatenar o valor e o índice, obtém-se uma característica que irá compor o vetor de características. E, finalmente, este vetor de características é concebido em cada um dos sensores.

Após produzir o vetor de características, o nó transmissor gera um polinômio aleatório de ordem N, previamente conhecido entre os dispositivos da rede. Os coeficientes desse polinômio são concatenados para formar a chave secreta utilizada na troca de mensagem entre os nós. A partir do vetor de características são constituídas as tuplas (pontos verdadeiros) que formam o cofre do PSKA. Também são geradas tuplas aleatórias, chamadas de pontos falsos. Esses pontos falsos permutados aleatoriamente com os pontos verdadeiros configuram o trancamento do cofre.

Para destrancar o cofre, o dispositivo receptor utilizará o vetor de características produzido por si próprio em conjunto com o cofre recebido do transmissor. É realizada uma operação de interseção entre o vetor e o cofre para extração dos pontos verdadeiros. A partir deles, reconstrói-se o polinômio original por meio da interpolação de pontos, no PSKA, é utilizada a interpolação de Lagrange. Com os coeficientes deste novo polinômio, a chave secreta é reconstruída e utilizada para a verificação do MAC (*Message Authentication Code*). Após o receptor destrancar o cofre com sucesso, é enviada uma mensagem com a chave para o nó transmissor e a autenticação é estabelecida.

B. OPFKA

O protocolo OPFKA [17], ilustrado pela Figura 2, possui um comportamento parecido com o protocolo PSKA, porém, se diferencia na forma de gerar as características e, consequentemente, no conteúdo do cofre. A geração do vetor de características se inicia com o cálculo de valores de intervalo interpulso (*IPI*) do sinal fisiológico aferido. Cada pico no sinal é detectado e o tempo até o próximo pico é quantificado e armazenado representando um *IPI*. Em sequência, os valores de *IPI* são representados de forma binárias e três *IPIs* adjacentes tem seus quatro bits menos significativos concatenados, formando uma característica de doze bits. Ao final, é realizada a operação de *hash* SHA-1 sobre cada um dos elementos resultantes. Os vinte primeiros bits do *hash* final representam uma característica a ser armazenada no cofre.

O cofre armazena as características verdadeiras em conjunto com características falsas, denominadas *ChaffPoints*. O cofre, no protocolo OPFKA, é trancado de modo similar ao protocolo PSKA, bastando apenas realizar permutações aleatórias dos elementos que o compõem. Portanto, as características legítimas ficam misturadas às características falsas.

O destrancamento do cofre pelo nó receptor é realizado após a extração de características equivalente a realizada no nó transmissor. No entanto, é realizada uma operação de identificação das posições com características legítimas no cofre recebido. Em seguida, uma mensagem é enviada para o nó transmissor contendo uma lista com os índices das posições encontradas. Nessa etapa, o nó transmissor é encarregado de verificar se as posições informadas são de fato referentes a características legítimas. No caso da quantidade das características indicadas de maneira correta for igual ou superior a um limite pré estabelecido, o acordo é realizado enviando uma mensagem de confirmação (MAC).

C. EKA

A técnica *EKG-Based* se diferencia do *Fuzzy Vault* pelo fato da chave comum entre o transmissor e o receptor ser constituída por sinais fisiológicos, especificamente o eletrocardiograma. Nos demais protocolos aqui citados, os sinais fisiológicos são utilizados para realizar o trancamento do cofre. Tratando-se do EKA [12], os dispositivos trocam os vetores de característica entre si e geram uma chave comum a partir de uma matriz de distância entre o vetor do transmissor e o vetor do receptor, ilustrado na Figura 3.

A constituição do vetor de características é similar ao PSKA, até a etapa de transformação do sinal do domínio de tempo para frequência. Após a aplicação da FFT pelo EKA,

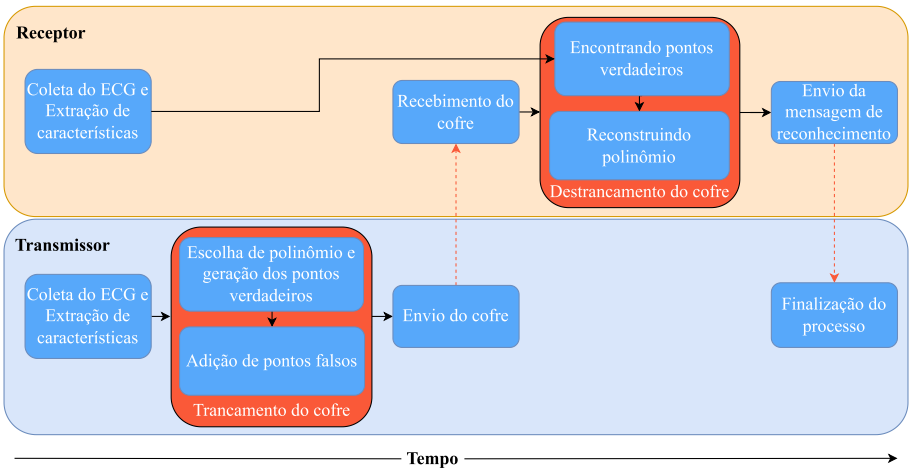


Figura 1: Protocolo PSKA

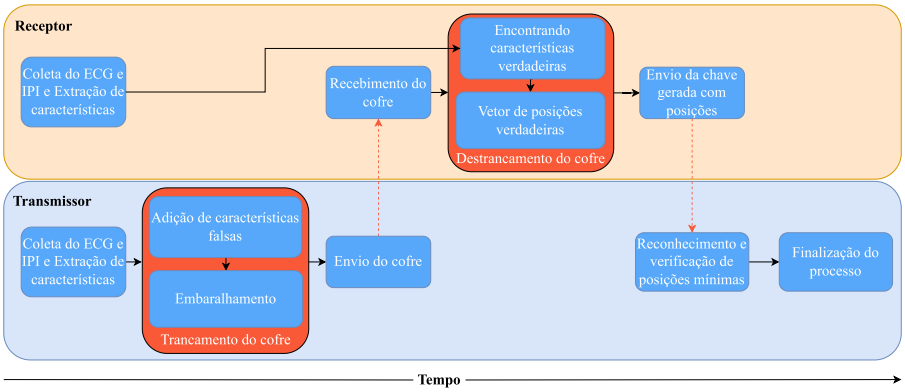


Figura 2: Protocolo OPFKA

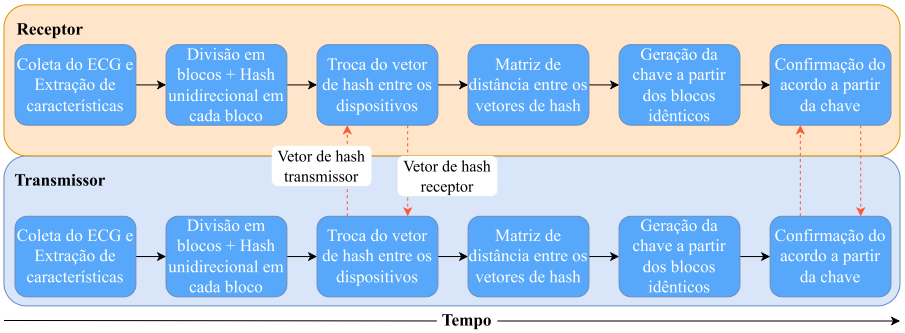


Figura 3: Protocolo EKA

ocorre a concatenação dos primeiros 64 coeficientes de cada janela, produzindo o vetor de características. Este vetor é posteriormente dividido em 20 blocos, com 16 coeficientes cada. Cada coeficiente é quantizado, resultando em um conjunto de representação binária de quatro bits por coeficiente. O resultado desta operação produz 20 blocos de 64 bits cada.

Para que ocorra a fase de compromisso entre os dispositivos comunicantes, cada um de seus respectivos blocos são preparados com aplicação de um *hash* unidirecional. Em seguida, tais blocos são trocados entre os dispositivos.

Subsequentemente, inicia-se a fase de processamento. Essa etapa identifica os blocos iguais em ambos os nós e produz a chave comum a partir deles. Para realizar a identificação, uma matriz de distância de *Hamming* é calculada entre os *hashes* dos blocos receptor e nó transmissor. Os índices da matriz indicarão quais os blocos que geraram os *hashes*. Com base nos índices, serão selecionados os blocos que tem o *hash* idêntico, ou seja, distância zero. Os blocos idênticos são agrupados em uma lista, a qual é submetida a função de *hash*. Assim é produzida a chave comum entre os dispositivos.

O descompromisso é a fase final do procedimento de autenticação. Nela ocorre a legitimação dos blocos enviados

durante o processo de compromisso. Para que isso aconteça, os dispositivos comunicam entre si enviando uma disjunção exclusiva entre a chave comum e a chave aleatória obtida na fase de compromisso. Após o envio da mensagem, o MAC é verificado utilizando a chave comum extraída dele. Se a verificação for um sucesso, o nó vai obter a chave aleatória produzida anteriormente manipulando a chave comum e a operação XOR. Caso a chave esteja correta, a verificação foi concluída e a autenticação realizada.

IV. Metodologia

Nesta Seção são apresentados os materiais e métodos utilizados para realizar a análise empírica dos protocolos. A base de dados designada para fornecer os sinais fisiológicos é pertencente a *Lobachevsky University Electrocardiography Database* (LUDB) [18]. Esta base é pública e mantida sob domínio do *Massachusetts Institute of Technology* (MIT), e pode ser obtida em Physiobank [3]. A LUDB, contém 200 registros de ECG, coletados de voluntários saudáveis e de pacientes do Hospital Nizhny Novgorod City entre 2017 e 2018. Estes registros são compostos por 10 segundos de aferições à 500 Hz. Nesta avaliação foram considerados 50 registros extraídos aleatoriamente da base para cada amostra experimental considerando a distribuição estatística normal.

Os protocolos foram implementados utilizando a linguagem Python 3.8. Todos os parâmetros e configurações das implementações foram ajustados de modo a reproduzir fielmente as propostas [12], [16], [17] e seus respectivos resultados. Um exemplo é a quantidade de "janelas"(seções) em que as amostras são divididas para extração de características, o qual foi definido como oito. Outro parâmetro, a ordem do polinômio utilizado para trancar o cofre no protocolo PSKA, foi definida como oito. No protocolo OPFKA foi definido tamanho de cofre igual a 300, quantidade de características verdadeiras igual a 12 e valor mínimo de características reconhecidas necessárias para ocorrer acordo igual a 10. E no EKA existe a parametrização da quantidade de blocos em que os coeficientes serão subdivididos para executar a fase de compromisso, a qual foi definida como 20 blocos.

Para validação das implementações foram realizados testes de FRR e FAR. Para os testes de falsa rejeição foram realizadas 100 repetições de acordo de chave válidos. Deste experimento, foi obtido a quantidade de rejeições em relação a quantidade de acordos que deveriam ser aceitos, resultando na métrica FRR. Para testes de falsa aceitação (FAR) em procedimento equivalente, foram contabilizados a quantidade de vezes que um acordo foi aceito quando deveria ser rejeitado.

A análise de desempenho avalia as etapas essenciais de cada protocolo, com objetivo proporcionar de forma empírica uma perspectiva do consumo de recursos durante cada fase de execução. Para representar o tempo consumido em cada etapa essencial dos protocolos, foi calculada a média e desvio padrão do tempo de execução (em milissegundos). Além do tempo, o consumo de memória RAM também foi aferido com o auxílio da biblioteca *tracemalloc* do Python. Isto permite quantificar os blocos de memória consumidos em cada etapa. Ademais, a análise de *goodput* foi realizada a fim de medir a quantidade de acordos que podem ser computados por segundo. A taxa de transmissão de dados (*bit rate*) foi estipulada conforme o desempenho alcançado em [19] (11.11 Kbps). Os experimentos foram simulados utilizando um Raspberry Pi 3 modelo B+ com sistema operacional Pi OS 1.7.1. Embora tenha poder computacional relativamente superior à dispositivos implantáveis, o Raspberry proporciona um ambiente de desenvolvimento adequado para emular de dispositivos vestíveis. Ele possui requisitos suficientes para coletar, armazenar e processar os dados referentes aos sinais biométricos. Ademais, permite avaliar, monitorar e explorar as especifidades dos softwares avaliados.

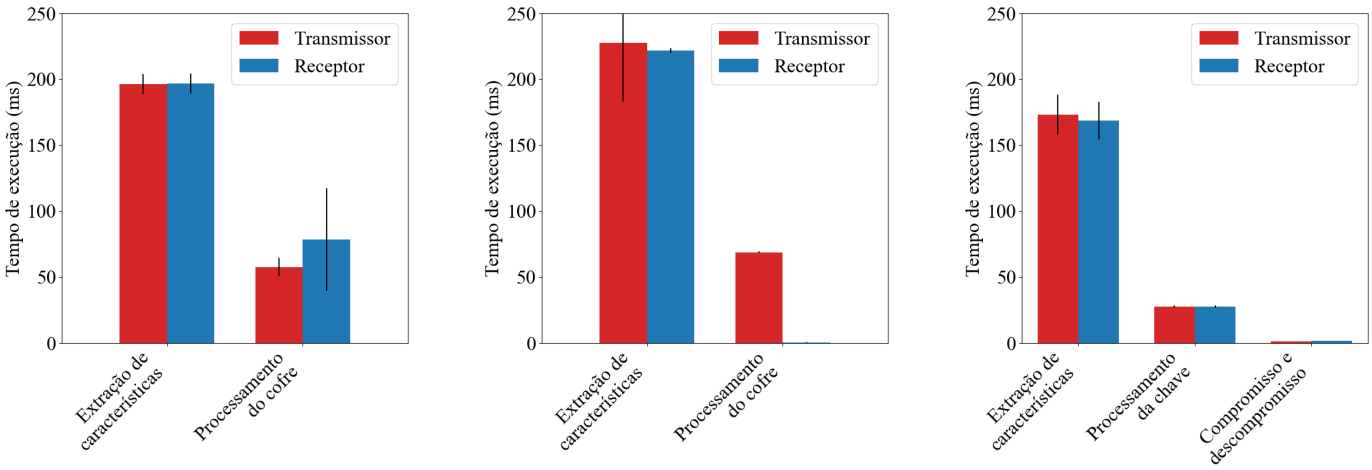
[3]https://www.physionet.org/content/ludb/1.0.0/

V. Resultados

Nesta Seção serão apontados e discutidos os resultados obtidos perante a comparação entre os protocolos PSKA, OPFKA e EKA. As métricas apresentadas são, o consumo de memória, tempo de execução, *goodput*, FRR e FAR. As Figuras 4 e 5, ilustram informações sobre tempo consumido por cada protocolo considerando dispositivo transmissor e receptor. Na Figura 4, são destacadas o consumo em cada fase do protocolo. E a Figura 5, destaca-se valores de consumo médio ($\bar{x}$) e desvio padrão (σ), comparando o tempo total de cada protocolo.

A Figura 4a, ilustras as duas fases. Primeiramente, a extração da característica, a qual engloba os procedimentos desde a coleta de amostras até a geração do vetor de características. O protocolo PSKA alcançou em média $\bar{x}= 196.51ms$ com um desvio de $\sigma = 7.6ms$ para o dispositivo transmissor. No receptor a média foi de $\bar{x} = 196.95ms$ e desvio de $\sigma = 7.62ms$. A outra etapa consiste na formação do cofre, o qual compreende a geração do polinômio e o trancamento do cofre no transmissor. Nesta fase o PSKA obteve um consumo médio de $\bar{x} = 57.75ms$ e desvio de $\sigma = 6.9ms$ para o dispositivo transmissor. No receptor, a etapa equivalente indica o destrancamento do cofre, com custo médio de $\bar{x} = 78.54ms$ e desvio de $\sigma = 38.93ms$. De forma geral, a Figura 4a, ilustra que em ambos transmissor e receptor, o tempo de extrair as características são equivalentes uma vez que processam o mesmo conjunto de funções. Entretanto, na etapa do processamento do cofre, destaca-se o maior consumo de tempo para o dispositivo receptor, 36% maior que o transmissor. Essa disparidade entre custo de tempo ocorre devido ao fato da operação de destrancamento do cofre exigir o processamento de funções adicionais para realizar reconstrução do polinômio. O desvio padrão elevado por parte do receptor indica que a quantidade de pontos variáveis na construção do cofre tem impacto direto sobre o tempo computacional.

A Figura 4b, também ilustra as mesmas duas fases. No protocolo OPFKA, a fase de extração de características engloba os procedimentos relacionados à leitura de sinais de ECG e cálculo dos intervalos interpulso (*IPIs*), além da criação das características através dos dois itens anteriores. O protocolo



(a) PSKA (b) OPFKA (c) EKA

Figura 4: Consumo médio de tempo.

OPFKA alcançou em média $\bar{x}$= $227.87ms$ com um desvio de $\sigma = 44.74ms$ para o dispositivo transmissor. No receptor a média foi de $\bar{x} = 221.99ms$ e desvio de $\sigma = 1.79ms$. A segunda etapa consiste no processamento do cofre, o que compreende a geração de características falsas e posteriormente um embaralhamento com características verdadeiras no transmissor. Por outro lado, no receptor, esta etapa consiste em localizar as posições das características verdadeiras no cofre e enviar de volta um vetor com tais posições para o transmissor. Nesta fase, o protocolo OPFKA obteve um consumo médio de $\bar{x} = 68.95ms$ e desvio de $\sigma = 0.57ms$ para o dispositivo transmissor. Já no receptor, a etapa apresentou custo médio de $\bar{x} = 0.64ms$ e desvio de $\sigma = 0.04ms$. Assim como no protocolo PSKA, a extração de características ocorre de maneira muito semelhante em transmissor e receptor no protocolo OPFKA. Dessa forma, o tempo necessário para essa etapa é muito próximo em ambos os nós. Porém, na etapa seguinte, responsável pelo processamento do cofre, o transmissor apresentou um maior consumo de tempo de execução. Isso acontece devido a cada nó realizar operações completamente diferentes nesta etapa.

Por pertencer a outra classe, o protocolo EKA possui além da etapa de extração de características, fases distintas dos demais protocolos. São elas, processamento da chave e compromisso/descompromisso, ilustradas na Figura 4c. Durante a extração de característica o consumo de tempo médio foi de $\bar{x} = 173.01ms$ com desvio de $\sigma = 15.17ms$ para o dispositivo transmissor e média $\bar{x} = 168.64ms$ com desvio de $\sigma = 14.27ms$ no receptor. A fase de compromisso e descompromisso engloba a geração das matrizes de *hashes* com custo médio de $\bar{x} = 1.43ms$ e desvio de $\sigma = 0.03ms$ para o transmissor e média $\bar{x} = 1.76ms$ com desvio de $\sigma = 0.03ms$ para o receptor. Já a fase de processamento da chave, a qual computa a matriz de distâncias e extrai a chave secreta, possui tempo médio de $\bar{x} = 27.73ms$ com desvio de $\sigma = 0.78ms$ no dispositivo transmissor, além do consumo médio de $\bar{x} = 27.63ms$ com desvio de $\sigma = 0.78ms$ no receptor. Ambos dispositivos (transmissor/receptor) possuem consumo de tempo equivalente perante em cada fase.

O consumo total de tempo dos protocolos é ilustrado na Figura 5. Observa-se que o protocolo PSKA tem um tempo maior de execução, em relação ao EKA, possui cerca de 25.8% a mais no transmissor e 39.1% no receptor. Por outro lado, em relação ao OPFKA, possui cerca de 16.8% a menos no transmissor e 23.7% a mais no receptor. O protocolo EKA apresentou o menor tempo de execução dentre os protocolos comparados. Isso ocorre devido aos protocolos PSKA e OPFKA realizarem operações mais custosas em cima dos dados coletados para realizar o acordo entre os nós. Já o protocolo EKA trabalha com o processamento da chave através de buscas em uma matriz de distância com número de blocos de tamanho fixo igual a 20. Em média, o tempo de execução do transmissor é 50 ms maior no PSKA do que no EKA. Considerando uma rede de vinte nós com atualização rotineira esta diferença resulta em aproximadamente um segundo, portanto, passível de causar conflitos para alguns nós durante o processo de acordo de chaves.

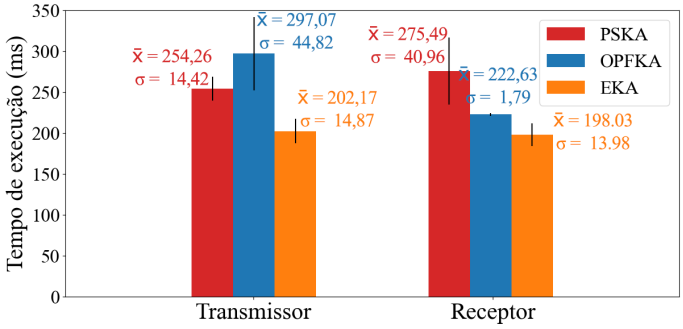


Figura 5: Tempo médio de execução (PSKA/OPFKA/EKA)

O consumo de médio de memória dos protocolos é ilustrado

pela Figura 6 (consumo por etapa) e Figura 7 (consumo total). A Figura 6a mostra que o protocolo PSKA consome em média $\bar{x} = 135.61KB$ de memória com desvio de $\sigma = 0.7KB$ na fase de extração de características em ambos os dispositivos. Na fase de processamento do cofre, o consumo médio foi $\bar{x} = 16.45KB$ com desvio de $\sigma = 1.71KB$ para o dispositivo transmissor e $\bar{x} = 12.35KB$ com desvio de $\sigma = 1.28KB$ para o dispositivo receptor. Esta variação de 33.2% ocorre devido ao fato de que o dispositivo transmissor armazena o polinômio gerado antes da construção do cofre. A Figura 6b ilustra o consumo de memória no protocolo OPFKA. Na Figura, o consumo de memória foi idêntico em ambos nós durante a fase de extração de características, com média de $\bar{x} = 111.36KB$ e desvio de $\sigma = 0.04KB$. Já na fase de processamento do cofre, o consumo médio foi $\bar{x} = 15.21KB$ com desvio de $\sigma = 0.04KB$ para o dispositivo transmissor e $\bar{x} = 0.23KB$ com desvio de $\sigma = 0.02KB$ para o dispositivo receptor. Tal diferença se origina devido ao receptor apenas receber o cofre e em seguida enviar um vetor com as posições de características verdadeiras para o receptor. Os resultados obtidos com o EKA foram idênticos em ambos dispositivos, ilustrados na Figura 6c, sendo a média $\bar{x} = 144.0KB$ com desvio de $\sigma = 0.0KB$ na fase de extração de características, $\bar{x} = 10.52KB$ com desvio de $\sigma = 0.0KB$ na fase de processamento da chave e $\bar{x} = 6.11KB$ com desvio de $\sigma = 0.0KB$ na fase compromisso e descompromisso.

Observando a Figura 7, percebe-se que o consumo de memória do EKA é maior que o PSKA, sendo 6% a mais no transmissor e 8.8% a mais no receptor. Contudo, o consumo de memória no OPFKA é inferior a ambos. Em relação ao PSKA, o OPFKA consome cerca de 18.6% menos no transmissor e 32.6% menos no receptor. Desta forma nota-se um *trade-off* entre tempo de execução e consumo de memória entre os protocolos. O PSKA e OPFKA lidam com operações de maior complexidade computacional para o processamento das características e geração do cofre. O EKA emprega menor processamento computacional sobre as características, entretanto consome mais memória devido a utilização de matrizes.

A análise da FAR e a FRR podem ser contempladas na Tabela I. A taxa de aceitação (AR) indica a porcentagem dos acordos que foram aceitos, ao somar AR com FRR, obtém-se o percentual que deveria ser aceito. Isso equivale também à taxa de rejeição (RR). Observa-se que o protocolo PSKA dispõe de uma taxa de falsa rejeição de 6,92% e falsa aceitação em torno de 11,3%. A taxa de falsa rejeição cresce de acordo com o tamanho do polinômio e em contrapartida a taxa de falsa aceitação diminui quando o tamanho do polinômio é maior. Além disso, o PSKA dispõe da possibilidade de aceitar uma requisição mesmo que os dispositivos não estejam completamente sincronizados. Ao avaliar o OPFKA e EKA, percebe-se que ambos atingem taxas incomuns de 0% de falsa aceitação e falsa rejeição. Isto ocorre devido a igualdade dos dados entre o receptor e o transmissor, pertinente a sincronização apurada.

É importante apontar que o PSKA não atinge o 0% de falsa rejeição ou falsa aceitação com a configuração que foi imposta, mesmo que os dados sejam iguais. Isso se deve ao fato de que as constantes que compõem o polinômio são geradas de forma aleatória e desta forma, ao realizar a interpolação dos pontos verdadeiros do cofre, o erro produzido pelo cálculo pode conduzir a uma falsa rejeição ou falsa aceitação. Para o cálculo do *goodput* foi considerado que cada protocolo envia uma quantidade fixa de bits, cerca de 6816 para o PSKA, 6992 para o OPKFA e 5632 para o EKA. Dessa forma, o *goodput* alcançado com as mensagens por cada protocolo, em acordos por segundo, foi de 1.63 para o PSKA, 1.59 para o OPFKA e 1.97 para o EKA.

Tabela I: Análise de confiabilidade do vetor de características (FAR/FRR)

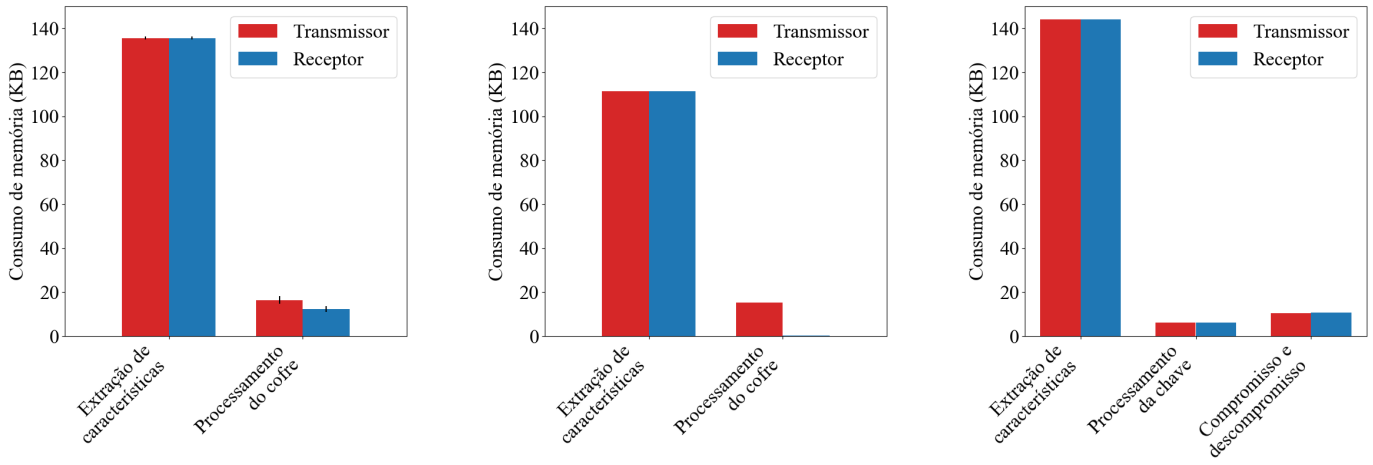
	AR	FRR	RR	FAR
PSKA	93.08%	6.92%	88.70%	11.30%
OPFKA	100%	0%	100%	0%
EKA	100%	0%	100%	0%

Em resumo, o protocolo OPFKA apresentou um tempo de execução mais longos que os demais, considerando o dispositivo transmissor. Para o receptor, o PSKA necessita de mais tempo. Em contrapartida, o protocolo OPFKA consome menos memória que os demais. Além disso, a análise de *goodput* indica que o OPFKA realiza menos acordos por segundo devido ao tamanho da mensagem a ser transmitida. Apesar dessas diferenças, a análise de FAR e FRR demonstrou que o EKA é inflexível com relação à sincronização dos sensores. Portanto, mais adequado quando considerados o uso de biometrias invariáveis como digitais e íris. Consequentemente, o PSKA é elegível a aplicações em ambiente real que consideram ECG e PPG como sinais biométricos devido à maior flexibilidade quanto a sincronização.

VI. Conclusão

Neste trabalho foi apresentado uma avaliação empírica de protocolos de autenticação de usuários para redes corporais, a qual compreendeu tempo de execução, consumo de memória e *goodput*. A avaliação procedeu sobre as principais etapas do processo de autenticação dos protocolos. Destacam-se as etapas de extração de características e o processamento da chave ou cofre. Os resultados mostram que a etapa de extração de características possui maior consumo de memória e tempo de execução em relação às demais etapas. O PSKA apresentou uma média de tempo de execução maior do que o EKA, em contrapartida o EKA demonstrou um consumo de memória mais elevado, demonstrando um *trade-off* de desempenho e memória entre as estratégias utilizadas. Além disso, o PSKA e OPFKA apresentaram um *goodput* menos satisfatório em relação ao EKA devido ao tamanho do cofre gerado. O tamanho do cofre nesse caso tem impacto direto no tamanho da mensagem de acordo de chaves.

Avaliando as métricas de falsa rejeição e falsa aceitação é possível afirmar que o OPFKA e EKA possuem melhor desempenho. Entretanto, exigem perfeita sincronização durante a coleta dos dados. Consequentemente é melhor aplicado a biometrias estáveis, tais como, digitais do dedo e íris. Portanto, o PSKA é passível de aplicação em cenário real considerando



(a) PSKA (b) OPFKA (c) EKA

Figura 6: Consumo médio de memória.

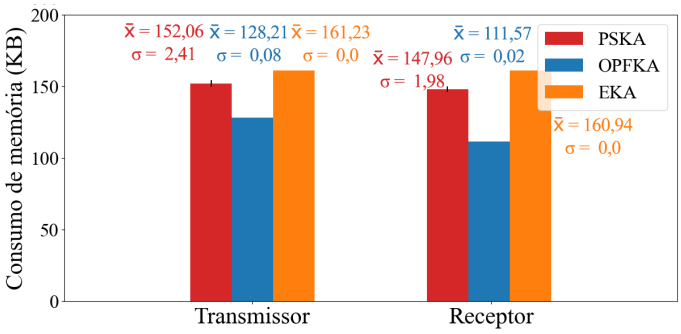


Figura 7: Consumo médio de memória (PSKA/OPFKA/EKA)

o PPG ou ECG como sinais biométricos, uma vez que, não exige uma sincronização com alto nível de precisão. Em trabalhos futuros, pretende-se desenvolver soluções para reduzir o consumo de tempo e memória durante a etapa de extração de características, a mais crítica do procedimento. Além disso, é desejável aprimorar soluções para lidar com dispositivos não sincronizados (OPFKA/EKA).

Referências

[1] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.

[2] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless body area networks: A survey," *IEEE Communications surveys & tutorials*, vol. 16, no. 3, pp. 1658–1686, 2014.

[3] Q. Liu, K. G. Mkongwa, and C. Zhang, "Performance issues in wireless body area networks for the healthcare application: A survey and future prospects," *SN Applied Sciences*, vol. 3, no. 2, pp. 1–19, 2021.

[4] K. K. Coelho, M. Nogueira, M. C. Marim, E. F. Silva, A. B. Vieira, and J. A. M. Nacif, "Lorena: Low memory symmetric-key generation method for based on group cryptography protocol applied to the internet of healthcare things," *IEEE Access*, vol. 10, pp. 12 564–12 579, 2022.

[5] S. Cherukuri, K. K. Venkatasubramanian, and S. K. S. Gupta, "Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in *2003 International Conference on Parallel Processing Workshops, 2003. Proceedings.*, 2003, pp. 432–439.

[6] A. Ali and F. A. Khan, "Key agreement schemes in wireless body area networks: Taxonomy and state-of-the-art," *Journal of medical systems*, vol. 39, no. 10, p. 115, 2015.

[7] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the internet of things," *Ad Hoc Networks*, vol. 32, pp. 17–31, 2015.

[8] A. V. Guglielmi, A. Muraro, G. Cisotto, and N. Laurenti, "Information theoretic key agreement protocol based on ecg signals," *arXiv preprint arXiv:2105.07037*, 2021.

[9] E. Marin, E. A. Rúa, D. Singelée, and B. Preneel, "A survey on physiological-signal-based security for medical devices," *Cryptology ePrint Archive*, 2016.

[10] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM conference on Computer and communications security*, 1999, pp. 28–36.

[11] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proceedings IEEE International Symposium on Information Theory,*, 2002, pp. 408–.

[12] K. K. Venkatasubramanian, A. Banerjee, and S. K. Gupta, "Ekg-based key agreement in body sensor networks," in *IEEE INFOCOM Workshops*. IEEE, 2008, pp. 1–6.

[13] H. Wang, H. Fang, L. Xing, and M. Chen, "An integrated biometric-based security framework using wavelet-domain hmm in wireless body area networks (wban)," in *2011 IEEE international conference on communications (ICC)*. IEEE, 2011, pp. 1–5.

[14] W. Wang, K. Hua, M. Hempel, D. Peng, H. Sharif, and H.-H. Chen, "A stochastic biometric authentication scheme using uniformed gmm in wireless body area sensor networks," in *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2010, pp. 1620–1624.

[15] C. C. Y. Poon, Yuan-Ting Zhang, and Shu-Di Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health," *IEEE Communications Magazine*, vol. 44, no. 4, 2006.

[16] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "Pska: Usable and secure key agreement scheme for body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 1, pp. 60–68, 2009.

[17] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen, "Opfka: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2274–2282.

[18] A. I. Kalyakulina, I. I. Yusipov, V. A. Moskalenko, A. V. Nikolskiy, A. A. Kozlov, K. A. Kosonogov, N. Y. Zolotykh, and M. V. Ivanchenko, "Lu electrocardiography database: a new open-access validation tool for delineation algorithms," *arXiv preprint arXiv:1809.03393*, 2018.

[19] A. Vale-Cardoso, M. Moreira, K. K. Coelho, A. Vieira, A. Santos, M. Nogueira, and J. A. M. Nacif, "A low-cost electronic system for human-body communication," *Electronics*, vol. 9, no. 11, p. 1928, 2020.