Universidade Federal do Rio de Janeiro
Instituto de Matemática
Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais
Programa de Pós-Graduação em Informática

Usando as propriedades físicas dos componentes dos medidores inteligentes para torná-los mais seguros

Alvaro Ernesto Robles Rincón

Rio de Janeiro
2022

Universidade Federal do Rio de Janeiro

Instituto de Matemática

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

Programa de Pós-Graduação em Informática

Alvaro Ernesto Robles Rincón

Usando as propriedades físicas dos componentes dos medidores inteligentes para torná-los mais seguros

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacitti da Universidade Federal do Rio de Janeiro (área de concentração: Sistemas Complexos Adaptativos), como parte dos requisitos necessários para a obtenção do Título de Doutor em Informática.

Orientador: Luiz Fernando Rust da Costa Carmo, Dr. UPS.

Coorientador: Wilson S. Melo Jr, DSc.

Rio de Janeiro

2022

CIP - Catalogação na Publicação

Rincón, Alvaro Ernesto Robles
R579u Usando as propriedades físicas dos componentes dos medidores inteligentes para torná-los mais seguros / Alvaro Ernesto Robles Rincón. -- Rio de Janeiro, 2022.
117 f.

Orientador: Luiz Fernando Rust da Costa Carmo.
Coorientador: Wilson de Souza Melo Junior.
Tese (doutorado) - Universidade Federal do Rio de Janeiro, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em informática, 2022.

1. Identidade. 2. Contexto histórico. 3. Características físicas. 4. Physical Unclonable Functions. 5. Medidores inteligentes. I. Carmo, Luiz Fernando Rust da Costa, orient. II. Melo Junior, Wilson de Souza, coorient. III. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

Usando as propriedades físicas dos componentes dos medidores inteligentes para torná-los mais seguros

Alvaro Ernesto Robles Rincón

Tese de Doutorado submetida ao Programa de Pós-graduação em Informática do Instituto de Matemática e do Instituo Tércio Pacitti da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários à obtenção do título de Doutor em Informática.

Aprovada em 16 de Agosto de 2022 por:

Prof. Luiz Fernando Rust da Costa Carmo, Dr. UPS. (Presidente)

participação por videoconferência

Prof. Wilson de Souza Melo Junior, D.Sc. (Coorientador), Inmetro

participação por videoconferência

Prof. Josefino Cabral Melo Lima, Docteur UPMC, UFRJ

participação por videoconferência

Prof. Claudio Miceli de Farias, D.Sc., UFRJ

participação por videoconferência

Prof. Davidson Rodrigo Boccardo, D.Sc, Greenhat

participação por videoconferência

Prof. Charles Bezerra do Prado, D.Sc, Inmetro

Para mis padres, mis hermanas, mi hermano y todas las personas que siempre me apoyaron y creyeron en mi.

AGRADECIMENTOS

Chegou o fim de um projeto acadêmico, parte de um percurso de vida, onde conheci várias pessoas e porque não dizer personagens. São tantos "gracias" e "muito obrigado" que seria impossível listar todos. No entanto, vou me arriscar e citar alguns.

Agradeço aos meus orientadores, Prof. Rust, pelo apoio incondicional e por me permitirem entrar neste mundo acadêmico, guiando-me durante todos estes anos, sem o Prof. Rust este caminho nem teria começado, obrigado por confiar em mim. Ao meu co-orientador Prof. Wilson pelo seu tempo e disponibilidade para me ajudar sempre que precisei, por acreditar em mim, me apoiar e me inspirar a ser cada vez melhor, sei que não foi uma tarefa fácil, aprendi muito, muito obrigado. Também quero agradecer a banca pelos comentários e recomendações realizadas, foram muito importantes.

Gostaria de agradecer ao programa de Doutorado Acadêmico e Inovação (DAI) da UFRJ em parceria com o Parque Tecnológico, Neopath e CNPq por me dar a oportunidade de fazer parte deste excelente projeto que incentiva a união da área acadêmica e das empresas. Eles foram uma parte crucial para terminar meu doutorado.

Aos meus colegas do laboratório Labnet, onde passei tantas horas e recebi grande apoio e carinho. Excelentes lembranças ficarão na minha memória, estou feliz por fazer parte do Labnet. Quero agradecer especialmente ao professor Claudio Micelli, um grande profissional e acima de tudo um excelente ser humano. Vou citar alguns amigos e colegas, espero não esquecer de ninguém. Aos Gabrieles (G1 e G2), ao Marcos (o louco do Pará), a Manú, ao Igor, ao Victor, à Beatriz, ao Hugo, à Jéssica, ao Felipe, ao Gustavo, ao Bruno, ao Ilan, à Vitória, Yago, Acacia e muitos outros, desculpe não citar todos. Mas acreditem, tenho boas lembranças de cada um de vocês.

No puede faltar el pilar en mis caminos de la vida, mi familia. Desde la distancia he sentido el apoyo, quiero agradecer especialmente a mi mamá, por nunca desistir ante la cantidad de adversidades que surgieron, con grande paciencia y amor me fue mostrando el camino desde niño. A mi papá que me enseñó a ver la vida a través de diversas perspectivas, gracias por los diversos consejos que me han servido en este camino que tomé. A mi hermana Jenny que ha sido un guía para todos nosotros, a mi hermana Laura, compañera de viajes y aventuras y mi hermano Ludwing que me enseña a ver la vida con gracia.

Também muito obrigado a você que leu todos os agradecimentos, e espero que esta tese seja interessante e útil.

Alvaro Ernesto Robles Rincón

Las cosas siempre pasan por algo...
(Desconocido)

RESUMO

Esta tese apresenta uma estratégia para aumentar a segurança de medidores inteligentes por meio das propriedades físicas, visando contornar ataques que comprometem componentes eletrônicos críticos. Muitos desses ataques tem o objetivo de obter ganho financeiro através da manipulação de medições de medidores inteligentes. Por exemplo, uma entidade maliciosa pode substituir ou adulterar componentes que executam procedimentos de medição e armazenam parâmetros sensíveis. Esse tipo de ataque é comum em dispositivos como medidores inteligentes utilizados em ambientes considerados hostis, onde os invasores têm fácil acesso ao dispositivo. Portanto, é necessário desenvolver técnicas nas quais as manipulações físicas em medidores inteligentes por invasores não sejam bem sucedidas. Nossa estratégia usa propriedades físicas desses componentes para criar identidades seguras para o medidor. Apresentamos duas contribuições principais. A primeira é inspirada em *Physical Unclonable Functions* e extrai identificadores exclusivos dos componentes dotados de memórias de programa (SRAM). Em seguida, combinamos esses identificadores para criar uma identidade forte. A outra contribuição usa informações de contexto físico dos níveis de tensão na fonte de alimentação do medidor inteligente para produzir identidades de contexto dinâmico. Também validamos nossas propostas em experimentos usando um protótipo de *hardware* que incorpora microprocessadores Arduino, memórias SRAM e sensores de tensão. Os resultados mostram que a estratégia desenvolvida é promissora para implementação em medidores inteligentes reais e pode ajudar a proteger esses dispositivos de ataques contra seus componentes.

Palavras-chaves: identidade; contexto físico; características físicas; *Physical Unclonable Functions*. medidores inteligentes.

ABSTRACT

This thesis presents a strategy to increase the security of smart meters through physical properties, aiming to circumvent attacks that compromise critical electronic components. Many of these attacks consist of getting undue advantages by manipulating smart meters' measurements. For example, a malicious entity can replace or manipulate components that perform measurement procedures and store sensitive parameters. This type of attack is common on devices such as smart meters used in environments considered to be hostile, where attackers have easy access to the device. Therefore, it is necessary to develop techniques where physical manipulations of smart meters by attackers are not successful. Our strategy uses physical properties from these components to create secure identities for the meter. We present two main contributions. The first one, is inspired by *Physical Unclonable Functions* and extracts unique identifiers from components equipped with program memories (SRAM). Then we combine these identifiers to create a strong identity. The other contribution uses physical context information from the voltage levels in the smart meter's power supply to yield dynamic context identities. We also validate our proposals in experiments using a hardware prototype that embeds Arduino microprocessors, SRAM memories, and voltage sensors. The results show that the developed strategy is promising for implementation in real smart meters and can help protect these devices from attacks against their components.

Keywords: identity; physical context; physical characteristics; Physical Unclonable Functions. smart meters.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ADC	Analog to Digital Converter
AHW	Assistidas por Hardware
ARM	Advanced RISC Machine
BP	BusPirate
CaF	Características Físicas
CAN	Controller Area Network
CF	Contexto Físico
CPS	Cyber Physical System
CPU	Central Processing Unit
DSP	Digital Signal Processor
DEK	Data Encryption Key
KEK	Key Encryption Key
DRAM	Dynamic Random Access Memory
ECDH	Elliptic-curve Diffie–Hellman
ECU	Engine Control Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
FAR	False Acception Rate
FRR	False Rejection Rate
GND	Ground
HD	Hamming Distance
IHM	Interface Homem-Maquina
IHMR	Interface Homem-Maquina e Rede
Inter-HD	Inter Hamming Distance

Intra-HD	Intra Hamming Distance
HT	Hardware Trojan
I2C	Inter-Integrated Circuit
INMETRO	Instituto Nacional de Metrologia, Qualidade e Tecnologia
IoT	Internet of Things
IPC	Inter-process Communication
JTAG	Joint Test Action Group
MAF	Filtros de Médias Móveis
MCU	Microcontrolador
MI	Medidores Inteligentes
ML	Metrologia Legal
MUX	Multiplexador
NIST	National Institute of Standards and Technology
NBTI	Negative Bias Temperature Instability
NFC	Near Field Communication
NLR	Não Legalmente Relevante
LCD	Liquid-crystal Display
LR	Legalmente Relevante
OIML	Organization Internationale de Métrologie Légale
OWASP	Open Web Application Security Project
PI	Propriedade Intelectual
PIN	Personal Identification Number
PUF	Physical Unclonable Functions
RO	Ring Oscillator
RTC	Real Time Clock

ROP	Return Oriented Programming
TCG	Trusted Computing Group
TPM	Trusted Platform Module
TRNGs	True Random Number Generators
SGX	Software Guard eXtensions
SPI	Serial Peripheral Interface
SoC	System on Chip
SRAM	Static Random Access Memory
USB	Universal Serial Bus
VCC	Voltage Commom Collector

SUMÁRIO

1 INTRODUÇÃO

Os avanços tecnológicos em *hardware* e *software* durante as últimas décadas permitiram e incentivaram a proliferação de dispositivos inteligentes integrados a diferentes sistemas físicos (BERTINO et al., 2016; QU et al., 2019). Atualmente é possível encontrar esses dispositivos em diferentes aplicações, tais como: sistemas autônomos de transporte (HUSSAIN; ZEADALLY, 2018; KIM et al., 2021), sistemas de *smart grid* (GHOSAL; CONTI, 2019), sistemas de monitoramento médico (XU et al., 2014; JAYATILLEKA; HALGAMUGE, 2020), sistemas de medição inteligentes (WERANGA; KUMARAWADU; CHANDIMA, 2014; WANG et al., 2019), entre outros. No entanto, essa proliferação trouxe um conjunto de desafios, principalmente aqueles relacionados à segurança da informação, em questões como identificação, autenticação, autorização e privacidade (WANG et al., 2016a; OKTAVIA et al., 2016; MOLLAH; AZAD; VASILAKOS, 2017; DABBAGH; RAYES, 2019). Esses desafios tornam-se mais complexos quando dois aspectos de um dispositivo são considerados: limites dos recursos de computação (SERPANOS; VOYIATZIS, 2013; RAGAB et al., 2019) e o ambiente operacional, especialmente quando esse ambiente é hostil (De Castro et al., 2017). Por ambiente hostil, consideramos locais onde entidades maliciosas tem acesso direto ao dispositivo (FOURNARIS; LAMPROPOULOS; KOUFOPAVLOU, 2017). Essas duas características frequentemente podem ser encontradas em dispositivos inteligentes e dispositivos IoT (Internet of Things) (ALLADI et al., 2020).

Considere um dispositivo inteligente localizado no ambiente hostil definido previamente, que precisa processar e enviar informações de forma segura. Em termos práticos, é desejável implementar criptografia baseada em chave pública para proteger tanto os elementos que armazenam dados sensíveis (por exemplo, bancos de memória) como também as interfaces de comunicação (YACCHIREMA; ESTEVE; PALAU, 2016; SRIDHAR; SMYS, 2017; RAGAB et al., 2019). De acordo com *National Institute of Standards and Technology* (NIST), a criptografia de chave pública é a mais encontrada nos sistemas atuais, e constitui uma forma segura de transferir e armazenar dados (BARKER; ROGINSKY, 2019). Porém, dispositivos mais simples, tais como IoT, sensores e medidores inteligentes, geralmente possuem limitações de memória, processamento e energização, o que pode inviabilizar o uso de aplicações criptográficas de chave pública, devido ao custo de: (i) criptografar e descriptografar as informações e (ii) gerenciamento de infraestrutura e distribuição de chaves, onde o número de chaves é diretamente proporcional ao número de dispositivos (GHALEB et al., 2016; SIDERIS et al., 2019). É importante ressaltar que, nestes casos, aprimorar a segurança de um dispositivo afeta seu desempenho e

custo final. Essa premissa é aplicável a qualquer dispositivo, inclusive os inteligentes (WURM et al., 2016). Portanto, é necessário encontrar um equilíbrio entre os recursos técnicos disponíveis em um dispositivo inteligente e os métodos de proteção aplicáveis a ele (GANGULY et al., 2016; SALLAM; BEHESHTI, 2019).

A literatura científica descreve um número crescente de ataques cibernéticos que tem como alvo dispositivos inteligentes ou mesmo componentes computacionais simples (WILLIAMS et al., 2017; PRINETTO; ROASCIO, 2020; CHEN et al., 2018a; BETTAYEB; NASIR; TALIB, 2019; WURM et al., 2016), cujos recursos disponíveis dificultam a implementação de mecanismos de segurança tradicionais como a criptografia (SARRAB; ALNAELI, 2019; Dhanesh Menon et al., 2019; ALLADI et al., 2020). Há poucos anos, a fabricante de veículos Jeep precisou fazer um *recall* de 1,4 milhões de unidades para trocar componentes físicos afetados por uma vulnerabilidade que permitia controlar o carro remotamente (MILLER, 2019). A vulnerabilidade explorava a conexão existente entre o centro de multimídia do veículo e seu processador central de bordo, chamado de *Engine Control Unit* (ECU). Em (SHOUKRY et al., 2013), os autores apresentam outro ataque que usa adulteração de componentes em ambientes automotivos. O ataque manipula as informações coletadas por um sensor encarregado de gerenciar o controle dos freios ABS de um carro antes de enviá-las ao ECU. Casos de ataques similares são reportados também na área da Metrologia Legal (Rodrigues Filho; GONÇALVES, 2015). Em (LEITÃO; VASCONCELLOS; BRANDÃO, 2014), os autores demonstram como a manipulação física dos componentes de *hardware* de bombas de combustível permite a adulteração das medições e, consequentemente, a alteração arbitrária do valor cobrado pelo combustível. Em geral, o atacante manipula o componente de medição encarregado de converter o fluxo de combustível em pulsos eletromagnéticos. Desta forma, a *Central Processing Unit* (CPU) da bomba de combustível utiliza os pulsos eletromagnéticos (manipulados) e realiza de forma indevida o cálculo do valor do combustível a ser cobrado do cliente. Além dos exemplos descritos, observa-se que existe uma preocupação crescente com a modificação de componentes de *hardware* em dispositivos inteligentes. Em (SHWARTZ et al., 2020), os autores levantam a questão de quão seguros são os componentes de *hardware* usados em reparos de *smartphones*. Por meio de um experimento, eles mostram que, através de um *touchscreen* modificado, é possível espiar o usuário e mesmo personificá-lo, entre muitas outras coisas.

Observa-se que os ataques apresentados possuem algumas características principais: (i) fraca ou nenhuma proteção contra manipulação física de componentes críticos e (ii) em caso de violação de um componente de medição, o sistema de controle desse dispositivo (e.g., *software* rodando na CPU) não rejeita o componente manipulado. É evidente que componentes que apresentem vulnerabilidades associadas a esses ataques

tornam-se um alvo a ser explorado, especialmente componentes que desempenham funções críticas, tais como medições de grandezas físicas ou ações de sensoriamento e controle. Portanto, surge um novo cenário em que dispositivos com recursos computacionais limitados precisam de novos mecanismos de proteção com confiabilidade equivalente àquela dos mecanismos encontrados em sistemas computacionais convencionais (LIU et al., 2017; GOYAL; DRAGONI; SPOGNARDI, 2016; BATINA et al., 2019).

1.1 MOTIVAÇÃO

Medidores inteligentes (MI) são uma importante classe de dispositivos IoT (Internet of Things) (SHROUF; ORDIERES; MIRAGLIOTTA, 2014) e podem ser descritos como blocos elementares na construção de sistemas ciberfísicos e de soluções para a Indústria 4.0 (KABALCI, 2016; Melo Jr. et al., 2019). Além disso, os medidores inteligentes constituem a nova geração dos instrumentos de medição responsáveis por regular as relações de consumo em uma sociedade moderna, assim como por estabelecer parâmetros de controle legal em atividades que envolvem negociação de bens mensuráveis, segurança, proteção à vida e conservação do meio ambiente (BOCCARDO et al., 2010; PRADO et al., 2014; Rodrigues Filho; GONÇALVES, 2015). Apenas na Europa, instrumentos de medição sob controle legal estão associados a um volume anual de negócios superior a 500 bilhões de euros (ESCHE; THIEL, 2015). Uma vez que a quase totalidade dos instrumentos de medição devem ser convertidos em medidores inteligentes nos próximos anos, a necessidade de garantir a confiabilidade e segurança cibernética destes dispositivos torna-se imperativa.

Uma parte significativa dos medidores inteligentes é constituída por dispositivos simples, de baixo custo, e com recursos computacionais limitados. Estas características são praticadas para evitar custos elevados da produção. Em muitos casos, o medidor inteligente pode ser reduzido a um simples sensor dotado de poder computacional suficiente para realizar cálculos de medição e transmitir as informações metrológicas para um equipamento coletor ou *gateway* (PETERS et al., 2015). Tais características de projeto podem tornar o medidor inteligente susceptível a ataques maliciosos (PRADO et al., 2014; ESCHE; THIEL, 2015; MELO et al., 2018). Se o medidor atua em um ambiente hostil, tais ataques podem incluir a manipulação física de seus componentes eletrônicos de forma maliciosa, o que inclui, por exemplo, a substituição de memórias que armazenam *software* ou parâmetros de calibração do instrumento (LEITÃO; VASCONCELLOS; BRANDÃO, 2014).

1.1.1 Estratégias de proteção de medidores inteligentes

Duas abordagens são aplicáveis para promover a segurança e confiabilidade de medidores inteligentes: abordagem baseadas em *software* (BOCCARDO et al., 2014; PETERS et al., 2015; De Castro et al., 2017; MELO et al., 2020); e abordagem baseada em *hardware* (NAGRA; COLLBERG, 2009; BRASSER et al., 2018).

Soluções de segurança baseadas em *software* fornecem (i) baixo custo de desenvolvimento e (ii) mínima ou nenhuma alteração física do dispositivo durante a implementação e futuras atualizações. São exemplos dessas abordagens a separação de *software* em funcionalidades legalmente relevantes (LR) e não legalmente relevantes (NLR) (BOCCARDO et al., 2014; Melo Jr. et al., 2019), o uso de arquiteturas virtualizadas para isolamento de funcionalidades (PETERS et al., 2015) e o controle de integridade de *software* por métodos de introspecção (De Castro et al., 2017). Soluções baseadas em *software* podem ser bastante eficientes para prevenir e detectar ataques contra instrumentos de medição, mas ao mesmo tempo demandam mais recursos computacionais, inclusive a implementação de aplicações criptográficas, sendo o uso de criptografia de chave pública em funcionalidades como assinatura e certificação digital amplamente recomendável (Melo Jr et al., 2020).

No caso das soluções baseadas em *hardware*, a implementação incorre na necessidade de desenvolver ou adquirir componentes específicos (CHEN et al., 2018b; BATINA et al., 2019). No entanto, talvez o ponto mais crítico para estas soluções seja aquele que envolve atualizações no dispositivo. Estas atualizações podem requerer a substituição de componentes ou, no pior dos casos, a substituição completa do dispositivo. Isso é mais agravante quando o dispositivo encontra-se no mercado, em outras palavras, quando o cliente já o possui. Os custos envolvidos para completar a atualização/substituição em centenas ou milhares de dispositivos podem ser tão altos que tornem inviável concluir o processo.

Outras propostas protegem componentes físicos de um medidor, como o microcontrolador e as memórias internas, para evitar alterações, adulterações e vazamentos de informação. Instrumentos de medição sob controle metrológico usualmente possuem lacres físicos usados para restringir o acesso ao seus componentes internos ou simplesmente evidenciar que houve um acesso (BOCCARDO et al., 2010; Rodrigues Filho; GONÇALVES, 2015). Todavia, tal mecanismo é frágil em cenários que dificultem a verificação do instrumento por uma terceira parte confiável (Melo Jr. et al., 2019). Propostas como *tamper proofing* (BOCCARDO et al., 2014; Melo Jr et al., 2020) ou *anti-tampering switches* (WARUDKAR; CHANDEL; SAWALE, 2014) ajudam a mitigar a manipulação de componentes por meio dos acessos ao interior do medidor. No entanto,

estas estratégias não garantem o estado dos componentes, pois os componentes podem ser violados sem restrições se o mecanismo de detecção de acesso falhar. Além disso, o *tamper proofing* pode ser contra efetivo uma vez que o componente praticamente se auto destrói impedindo uma fraude, mas ao mesmo tempo inutiliza suas funcionalidades.

Uma abordagem pouco explorada na proteção de medidores parte da ideia de se implementar mecanismos que identifiquem e monitorem os componentes de *hardware* de um dispositivo. Em termos práticos, tal estratégia pode ser descrita como híbrida, uma vez que sua implementação depende tanto de funcionalidades providas pelo *hardware* quanto pelo *software*. Basicamente, ela consiste em estabelecer identidades físicas de componentes sensíveis em um medidor, e monitorar a consistência dessas identidades em tempo de execução. Os conceitos teóricos que fundamentam essa estratégia são descritos a seguir.

1.1.2 Identificação, autenticação, identidade e características físicas

Identificação pode ser definida como o procedimento no qual uma entidade é identificada ou reconhecida por um sistema, e autenticação é a forma para se estabelecer a validade da identidade avaliada (STALLINGS, 2014). Por sua vez, a identidade consiste de um fragmento de informação que possa ser atribuído de forma unívoca à uma entidade, como por exemplo um código numérico, uma assinatura, ou uma biometria. Recentemente, há um crescente interesse em se obter identidades a partir de características físicas de dispositivos computacionais (TSOUTSOS; KONSTANTINOU; MANIATAKOS, 2014; XU; WENDT; POTKONJAK, 2014; GUIN; ASADIZANJANI; TEHRANIPOOR, 2019). Atualmente *Physical Unclonable Functions* (PUF) (GUAJARDO et al., 2007; ADAMES; DAS; BHANJA, 2016) é o método utilizado para obtenção de identificadores físicos de entidades tais como componentes eletrônicos.

Os PUFs exploram propriedades físicas adquiridas pelo componente durante o processo de fabricação, extraindo valores únicos que permitam identificá-lo (GUAJARDO et al., 2007). Como premissa, PUFs devem ser fáceis de se computar e avaliar, mas difíceis de prever, mesmo para dois componentes produzidos de forma idêntica (SAKHARE; SAKHARE, 2020). Isto permite considerar os PUFs como uma "impressão digital" para dispositivos (MIAO et al., 2016). De forma similar à biometria (RUI; YAN, 2018; BORRA; REDDY; REDDY, 2016), os valores obtidos dos PUFs podem ser aplicados em métodos de segurança como geração de identificadores e chaves criptográficas (Günlü et al., 2019), e também em métodos de controles de acesso envolvendo identificação e autenticação (AMAN; CHUA; SIKDAR, 2017; MUHAL et al., 2018). Na segurança física de um medidor inteligente, PUFs podem ser úteis para detectar a manipulação indevida de elementos de *hardware* como *chips* ou memórias (STANCIU; MOLDOVEANU; CIRSTEA,

2016; LEE; MARKANTONAKIS; AKRAM, 2016). Eles também podem ser usados como identificadores físicos de componentes internos do dispositivo, sendo todavia limitados àqueles componentes que permitem a extração e mapeamento de suas propriedades intrínsecas (*chips* e memórias) (BOYAPALLY et al., 2020).

Outra área do conhecimento que explora características físicas na construção de mecanismos de segurança para dispositivos ciberfísicos diz respeito ao uso de informações de Contexto Físico (CF) na geração de identidades e identificadores, que são determinados pelo ambiente onde o dispositivo está inserido, ou ainda por seu comportamento. Nos últimos anos, alguns trabalhos científicos exploraram essa estratégia para propor métodos e técnicas de segurança cibernética, com aplicações envolvendo comunicação entre sensores (ZENGER et al., 2015), autenticação de dispositivos IoT (FOMICHEV et al., 2019), e obtenção de canais seguros com base no contexto do ambiente (MELO et al., 2018). Embora o uso de características físicas e do contexto físico como método para gerar identificadores e identidades seja relativamente recente, esse método mostra-se promissor para a área de segurança da informação, e pode ser aplicado tanto em soluções de segurança baseadas em *software* como em *hardware*.

1.1.3 Desafios e oportunidades

Na subseção anterior, foi discutido como as características físicas e o contexto físico associados a um dispositivo inteligente podem ser usados em métodos e técnicas de identificação e autenticação. Especialmente em relação aos medidores inteligentes, embora tal abordagem seja bastante promissora, ela ainda é pouco difundida na literatura científica, e menos ainda em soluções práticas implementadas pela indústria. De fato, ainda existem lacunas de conhecimento a respeito de como manter um medidor inteligente seguro, principalmente quando se considera sua proteção contra manipulações físicas de componentes de *hardware*. Algumas dessas questões em aberto são:

- (i) ainda é difícil garantir a não violação e a troca de componentes internos em medidores inteligentes.
- (ii) poucos estudos exploram o contexto físico e as características físicas para evitar violação ou troca de componentes internos em medidores inteligentes.
- (iii) Poucos trabalhos consideram o uso de propriedades físicas dos componentes internos de medidores inteligentes para gerar identidades.

Técnicas com características físicas, como PUFs, possuem uma metodologia aprofundada e são uma forma promissora de gerar identidades físicas para dispositivos

computacionais, inclusive medidores inteligentes. Como a maioria das técnicas atuais para gerar identidades, aquelas baseadas em PUF focam sua metodologia em processadores e memórias (MAES; VERBAUWHEDE, 2010; ADAMES; DAS; BHANJA, 2016). Isso é esperado, uma vez que existem muitos componentes dentro de um dispositivo e não é possível proteger cada um deles. Em relação ao contexto físico, o conceito de se observar e monitorar eventos físicos esperados no ecossistema de funcionamento de um medidor inteligente pode permitir a detecção e bloqueio de modificações em componentes físicos não previstas, que por sua vez possivelmente estão associadas ou a falhas do equipamento ou a ataques maliciosos deliberados. Deste modo, entende-se que o estudo desses temas constitui uma oportunidade de pesquisa relevante, que pode por sua vez resultar em benefícios diretos para fabricantes de instrumentos de medição, usuários desses medidores, e para a sociedade como um todo de forma indireta.

1.2 OBJETIVOS DA TESE

O objetivo principal desta tese é investigar e propor soluções tecnológicas que, a partir de componentes internos de um medidor inteligente, permitam a obtenção de identidades físicas deste dispositivo, por meio da aplicação de metodologias que explorem características físicas e o contexto físico associados a estes componentes. O objetivo principal, por sua vez, se desdobra nos seguintes objetivos secundários que delimitam o escopo de pesquisa e desenvolvimento desta tese:

- Investigar a fundamentação teórica para o conceito de uso de propriedades físicas e contexto físico como métodos para gerar identidades de componentes.
- Propor metodologias que evidenciem o uso de características físicas e contexto físico para criar identidades em medidores inteligentes.
- Avaliar como as características intrínsecas obtidas dos componentes de um medidor inteligente permitem a obtenção de identidades.
- Implementar casos práticos das metodologias propostas, demostrando como através delas é possível ter um maior nível de proteção em medidores inteligentes.

1.3 PROPOSTA DE TESE

Para esta tese propomos explorar o conceito de identidade, para representar de forma única um dispositivo e os seus componentes. Esta identidade será aplicada a dispositivos que medem grandezas físicas, mais especificamente a medidores inteligentes. A identidade resultante do medidor inteligente será a conjunção de componentes internos que fazem parte do processo de medição.

Dois mecanismos foram propostos para criar identidades dos componentes para medidores inteligentes. A primeira é uma estratégia inspirada em *Physical Clonable Function* (PUF) (GUAJARDO et al., 2007; TUYLS, 2010; HERDER et al., 2014; ADAMES; DAS; BHANJA, 2016) para gerar identidades de dispositivos a partir de valores únicos extraídos de componentes passivos (por exemplo, memórias). Nós expandimos essa estratégia e propomos que componentes envolvidos no processo de medição em um medidor inteligente se comportem como um PUF. Portanto, a identidade é baseada no conjunto de componentes do medidor inteligente. Diferente das implementações atuais de identidades usando PUFs, onde identidade do dispositivo é baseada em um único componente (chip ou memória) (RÜHRMAIR; HOLCOMB, 2014; LEE; MARKANTONAKIS; AKRAM, 2016; ALLADI et al., 2020). Assim, caso exista a troca de algum componente de medição, a identidade daquele medidor inteligente será diferente da esperada e medidas de segurança serão tomadas. O PUF usado nos experimentos foram memórias *Static Random Access Memory* (SRAM) consideravelmente pequenas, onde através de uma analise estatística dos estados iniciais dos *bits*, foi possível extrair identidades dos componentes do medidor inteligente. Para validar as identidades obtidas, foram utilizadas métricas usadas em PUF (TEHRANIPOOR et al., 2017a; SURI; CHAKRABORTY, 2018), especificamente, Intra *Hamming Distance* (Intra-HD) e Inter *Hamming Distance* (Inter-HD). Por meio do Intra-HD, foram avaliadas as diferenças em *bits* de uma identidade obtida repetidamente de uma SRAM. Por sua vez, o Inter-HD permitiu determinar quão diferentes são as identidades obtidas entre memórias SRAM fisicamente idênticas. Os resultados dos experimentos realizados demonstra que a identidade resultante distingue efetivamente os componentes, com base em suas imperfeições construtivas intrínsecas.

O segundo mecanismo é inspirado no Contexto Físico (CF) gerado pelo comportamento interno do medidor inteligente. O CF pode ser entendido como o conjunto dos eventos físicos associados a uma ou mais grandezas físicas observáveis por sensores (MELO, 2018). Desse modo, nesta tese, desenvolvemos um método que obtém identidades dinâmicas baseadas no CF observável por componentes que compartilham a mesma fonte de alimentação comutada. Em outras palavras, propomos que flutuações na alimentação de energia dos componentes sejam o CF para um medidor inteligente. A proposta baseia-se no comportamento das fontes de alimentação que tentam continuamente estabilizar o nível de tensão, produzindo flutuações em função do consumo de energia dos componentes.

Uma característica importante para produzir identidades a partir de um CF é possuir unicidade e não ser previsível (MELO, 2018), neste sentido foi proposta a utilização das variações de tensão do medidor inteligente. Componentes que podem descrever esta variação provam que fazem parte do dispositivo e que compartilham a mesma fonte de alimentação. A avaliação das identidades foi realizada através das métricas

False Rejection Rate (FRR) e *False Acceptance Rate* (FAR), comumente utilizadas em identificação biométrica. Por meio dessas métricas é possível saber a aceitação ou negação que a identidade proposta pode ter em um sistema. Adicionalmente, experimentos foram realizados para conhecer o *threshold* (*th)*, mais especificamente o valor *th* que permite ter menor quantidade de FRR e FAR.

Ambas as estratégias de identidade foram implementadas usando um protótipo de *hardware* de medidor inteligente, incluindo componentes como microprocessadores Arduino, memórias SRAM, e sensores de tensão. Utilizamos este protótipo para obter resultados práticos que atestam a viabilidade de nossas estratégias de segurança.

1.4 COMO ESTA TESE ESTÁ ORGANIZADA

Capítulo 2 – Conceitos Básicos – Este capítulo apresenta os conceitos necessários para entender os trabalhos desenvolvidos nesta tese de doutorado. Os conceitos discutidos neste capítulo referem-se à identificação, autenticação e identidade. Posteriormente são definidos os conceitos de computação segura, suas aplicabilidades e soluções desenvolvidas. Também é apresentado o uso de características físicas como método de segurança. Finalmente, são apresentados tipos de vulnerabilidades encontradas em dispositivos inteligentes, com ênfase especial em ataques de nível *hardware*. Para os leitores que estão familiarizados com estes conceitos, este capítulo é opcional. Contudo, toda alusão nos subsequentes capítulos está devidamente referenciada.

Capítulo 3 - Trabalhos relacionados - Este capítulo apresenta as soluções existentes para o desenvolvimento de identidades em dispositivos inteligentes. As identidades apresentadas incluem aquelas baseadas em *hardware* e propriedades de *hardware*. Especificamente, são apresentadas soluções que utilizam componentes como TPM para computação segura, bem como o uso do contexto físico e das características físicas dos dispositivos inteligentes. Uma tabela de comparação também está incluída com as técnicas existentes e propostas.

Capítulo 4 – Identidades Físicas para Medidores Inteligentes – Este capítulo apresenta um método para aplicar segurança através das características físicas de um medidor inteligente. O método aproveita técnicas baseadas em *Physical Unclonable Function* (PUF) para gerar as identidades. Os experimentos foram desenvolvidos usando um microprocessador Arduino e as memórias SRAM nele inclusas.

Capítulo 5 - Identidades baseadas no contexto físico - Este capítulo apresenta uma nova forma de uso do contexto físico em dispositivos inteligentes, mais especificamente em medidores inteligentes. Nós apresentamos dois experimentos principais, o primeiro

sobre a validação do contexto físico proposto, o segundo sobre como um componente dentro do medidor inteligente pode identificar o contexto físico. No final avaliamos as propriedades de segurança das identidades geradas a partir do contexto físico.

Capítulo 6 - Conclusões e trabalhos futuros - Neste capítulo apresentamos os resultados finais desta tese. Explicamos como atingimos os objetivos propostos. As limitações e os pontos fortes das propostas desenvolvidas são explicados em detalhes. Da mesma forma, são apresentadas oportunidades de melhoria e, por fim, trabalhos futuros.

2 CONCEITOS BÁSICOS

Este capítulo se destina a apresentar conceitos preliminares que são prerrequisitos para a compreensão deste trabalho. Ele está organizado nas seguintes seções, as quais são independentes entre si em um primeiro momento. A primeira fala sobre Metrologia Legal e medidores inteligentes, incluindo integridade e fraude em processos de medição. Também falamos sobre vulnerabilidades em dispositivos inteligentes. O conceito de identidade e elementos relacionados, como identificação e autenticação, são discutidos. Também apresentamos o uso de características físicas e Contexto Físico (CF) como um método de proteção de dispositivos.

2.1 INTEGRIDADE DE COMPONENTES FÍSICOS EM DISPOSITIVOS INTELIGENTES

Os dispositivos inteligentes atualmente ocupam uma grande porcentagem dos dispositivos em todo o mundo (EBERT; JONES, 2009; AKDUR; GAROUSI; DEMIRÖRS, 2018). Sem dúvida, a proteção dos dispositivos inteligentes é um fator importante na segurança da informação, principalmente pelo fato deles estarem cada vez mais interconectados, por exemplo por meio da Internet (WURM et al., 2016; WILLIAMS et al., 2017; ALLADI et al., 2020). Outro aspecto a ser considerado é o fato de que podem ser encontrados em processos de medição, caso em que são chamados de instrumentos de medição ou medidores inteligentes. Basicamente, são dispositivos que possuem a capacidade de medir grandezas físicas como energia, temperatura, tempo, volume, velocidade, pressão e diversas outras grandezas. Aumentar a integridade de seus componentes de *hardware* e *software* é um desafio que requer melhorias em seu desenvolvimento, manutenção, inspeção e controle.

Garantir a precisão e confiabilidade das medições em medidores inteligentes é fundamental para qualquer sociedade, e a busca por práticas e mecanismos que promovam essas propriedades está em constante evolução (PRADO et al., 2014; PETERS et al., 2015; ESCHE; THIEL, 2015; Melo Jr et al., 2020). A confiabilidade da medição envolve vários fatores, incluindo o *software* e *hardware* responsável pela medição. Os institutos de metrologia em cada país são responsáveis por tomar ações para garantir essa confiabilidade, como por exemplo proceder com a aprovação de modelo (*type approval*) de um novo medidor inteligente, ou verificar sua conformidade antes e durante a utilização (*market/field surveillance*) (Rodrigues Filho; GONÇALVES, 2015). Entretanto, as estratégias de proteção de medidores inteligentes são mais efetivas quando implementadas em tempo de projeto, ou seja, o medidor precisa ser dotado de mecanismos confiáveis

que facilitem a detecção de ataques ou ações maliciosas, e possam tomar decisões que não comprometam o funcionamento correto do instrumento (PETERS et al., 2015; De Castro et al., 2017).

2.1.1 Metrologia Legal

A Metrologia Legal (ML) é uma área de conhecimento dentro da Metrologia relacionada às atividades que envolve medições, unidades de medida, instrumentos e métodos de medição. Em termos gerais, a ML pode ser definida como o campo da Metrologia que se preocupa com a confiabilidade dos instrumentos de medição usados nas relações de consumo, ou em atividades que impliquem proteção à vida (e.g., aplicações médicas) e ao meio ambiente. A ML é aplicável desde os processos de fabricação dos instrumentos de medição, e também quando o instrumento encontra-se no mercado (BOCCARDO et al., 2010; Rodrigues Filho; GONÇALVES, 2015). Dentre as vantagens decorrentes do controle legal metrológico dos instrumentos de medição destacam-se:

- Garantir a confiabilidade das medições dos instrumentos;
- Reduzir as perdas financeiras das empresas;
- Aumentar a competitividade do país;
- Reduzir a incerteza dos usuários sobre os instrumentos de medição.

Desta forma, a ML pode ser vista como um facilitador entre fornecedores de instrumentos de medição e usuários finais ou consumidores. A forma como as atividades de controle legal metrológico são implementadas é diferente em cada país, devido às suas características sociais e econômicas (MELO et al., 2018). Assim, cada país deve analisar e estudar quais medidas são aplicáveis a eles. No Brasil, a autoridade responsável pela condução da Metrologia Legal é o Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro), onde, por meio de um conjunto de regulamentos técnico metrológicos, estabelece os requisitos legais para dispositivos de medição.

2.1.2 Regulamentação e Controle Metrológico

A regulamentação, e consequentemente o controle metrológico dos instrumentos de medição, dependerão especificamente dos requisitos legais adotado por cada país, bem como da compreensão dos conceitos de Metrologia (Rodrigues Filho; GONÇALVES, 2015). Em (PETERS et al., 2015), os autores mostram como a regulamentação e o controle metrológico em instrumentos de medição tem impactos diretos na sociedade e na economia dos países.

Para ajudar no processo regulatório do Controle Metrológico, existem documentos internacionais que funcionam como diretivas para cada classe de instrumentos de medição. São exemplos desses documentos o OIML-D31 da *Organization Internationale de Métrologie Légale* (LÉGALE, 2008) e o WELMEC 7.2 da *Western European Legal Cooperation for Legal Metrology* (GUIDE, 2008), que tratam especificamente de equipamentos de medição controlados por *software*. Esses documentos são aceitos pela maioria dos países e definem aspectos dentro do escopo de atividades da ML, como: vocabulário, padrões de *software*, e inspeção de instrumentos de medição. Nessa ordem de ideias, cada país desenvolve seu próprio arcabouço legal e estabelecerá sua própria estrutura organizacional, incluindo pessoal especializado. Estas iniciativas usualmente são atreladas ao Instituto Nacional de Metrologia de cada país (PETERS et al., 2015). Usualmente, as atividades desenvolvidas pelo controle de metrológico incluem (Rodrigues Filho; GONÇALVES, 2015):

- Aprovação de modelo, que consiste na apreciação do projeto construtivo de um medidor, autorizando ou não a sua fabricação;
- Vigilância do sistema de qualidade (através de órgãos autorizados para autoverificação);
- Vigilância do mercado (avaliação do instrumento de medição antes de ser enviado para o mercado);
- Vigilância de campo (verificação do instrumento de medição em uso).

Em geral, a vigilância do sistema de qualidade e a vigilância do mercado buscam o controle legal, a aprovação, a validação e a verificação do tipo de instrumento de medição (MELO et al., 2018; Rodrigues Filho; GONÇALVES, 2015). Na vigilância do mercado, uma verificação é realizada nos instrumentos de medição antes de chegar ao consumidor. No caso da vigilância de campo, esta basicamente monitora o instrumento de medição, incluindo sua conformidade, no ambiente para o qual foi destinado(MELO et al., 2018). Juntas, essas atividades auxiliam na detecção de fraudes em instrumentos de medição por meio de ações preventivas e de supervisão.

2.1.3 Medidores inteligentes

Medidores inteligentes (MI) são dispositivos com capacidade de processar grandezas físicas e dados digitais complexos (PRADO et al., 2014). Existem vários tipos de medidores inteligentes, desde aqueles que medem o oxigênio no sangue, batimentos cardíacos, até os usados para identificar o consumo de serviços públicos como água,

gás e energia elétrica. Uma característica dos medidores inteligentes utilizados para identificar o consumo é estabelecer comunicação com a distribuidora em tempo real (ASGHAR et al., 2017). Quando há comunicação bidirecional, o provedor de serviços pode receber e enviar comandos diretamente para o medidor inteligente (WERANGA; KUMARAWADU; CHANDIMA, 2014). Entre as operações gerenciáveis nos medidores que identificam o consumo, encontram-se (PAUL et al., 2014):

- Definição parâmetros do medidor sob demanda de consumo;
- Controle distribuído;
- Detecção e diagnóstico de falhas;
- Armazenamento de dados;
- Inquérito de consumo;
- Geração de fatura;
- Comunicação ponta a ponta;
- Interface digital.

Nesse sentido, o distribuidor, além de gerenciar processos do MI, tem a capacidade de gerenciar ações dentro da rede. Assim, os MI podem ser definidos como dispositivos de comunicação de *endpoint*, que por sua vez, exigem considerações de segurança por estarem em locais potencialmente hostis (Dhanesh Menon et al., 2019).

2.1.4 Integridade em medidores inteligentes

Regulamentações e controles metrológicos são essenciais para a realização de ações que ajudem a manter a integridade do MI. A integridade visa garantir que o *software* e o *hardware* do MI permaneçam legítimos e inalterados. (PRADO et al., 2014). Duas estrategias são comumente elaboradas para manter a integridade:

- Analise do medidor: No *software* são analisadas as funções envolvidas nos processos de medição. No *hardware* são analisados os componentes e sensores envolvidos no processo de medição.
- Inspeção do medidor: quando um medidor está em campo, é necessário verificar se o mesmo não sofreu alterações que afetem seu desempenho de medição. A inspeção do medidor permite comparar a versão atual com uma versão previamente validada pelo fabricante e autoridades regulatórias (BOCCARDO et al., 2010).

Para facilitar os processos de aprovação de modelo e vigilância de campo de MI controlados por *software*, a Metrologia Legal traz o conceito de cadeia Legalmente Relevante (LR), que engloba todos os elementos de *software*, *hardware* e dados envolvidos na geração, processamento e exibição dos valores de medição (INMETRO, 2018). A cadeia LR, por sua vez, permite focar a analise e a verificação da integridade exclusivamente nos componentes (*software* e *hardware*) que afetam o resultado da medição. Incluindo os componentes que interagem direta ou indiretamente com o processador/memória.

Para analisar o *software* é possível usar varredores de classes e funções como Veracode (VERACODE, 2021) e Sonarqube (SONAQUBE, 2021). Ferramentas como IDA pro (EAGLE, 2011) permitem analisar o código binário do *software* do medidor, identificando caminhos de execução associados aos processos de cálculo de medições. Para garantir a integridade do *hardware* envolvido nas medições, são realizados testes com parâmetros que simulam o comportamento do medidor em campo.

Quando o dispositivo está em campo, a verificação pode ser realizada por meio de testes funcionais, como os realizados durante a aprovação. Para evitar aberturas não esperadas do medidor, são usadas estratégias de controle de manipulação do medidor, como o *anti-tampering*/blindagem (PALEY; HOQUE; BHUNIA, 2016). Também é possível realizar introspecção de *software* por meio de mecanismos de auditoria que detectam modificações inadvertidas (De Castro et al., 2017).

2.1.5 Fraude em processos de medição

Fraudes em medidores inteligentes podem ser extremamente lucrativas e, ao mesmo tempo, difícil de detectar (Melo Jr et al., 2019). As fraudes podem ter origem no *software* ou *hardware*, porém, as baseadas em *hardware* são as mais complexas de se identificar, mesmo durante as verificações metrológicas realizadas em campo (BOCCARDO et al., 2010; KONSTANTINOU; MANIATAKOS, 2019; PETERS et al., 2020). Nas fraudes que envolvem *hardware*, os componentes do medidor são modificados com o intuito de dificultar a detecção da fraude através de inspeção visual (LEITÃO; VASCONCELLOS; BRANDÃO, 2014). Para detectar a fraude são necessários testes de funcionalidade aplicados ao medidor. Estes testes consistem em comparar medições obtidas em campo com medições realizadas em ambientes controlados e seguros. Contudo, técnicas mais avançadas têm a capacidade de desencadear fraudes remotamente, contornando os testes de fiscalização realizados em campo (Melo Jr et al., 2019; LEITÃO; VASCONCELLOS; BRANDÃO, 2014). Assim, quando é realizada a fiscalização, a fraude é desativada, dando a impressão de que o medidor não foi adulterado.

Para contextualizar a fraude, considere o exemplo a seguir: um medidor

de uma bomba de combustível que obedece aos padrões estritos de segurança contra ataques de integridade tem um processador que usa uma estratégia baseada em PUF. Este processador gera uma identidade que é usada como chave para proteção dos dados na memória e do *software* através de ferramentas criptográficas. Porém, durante o uso normal, usuários descobrem discrepâncias na quantidade paga pela mesma quantidade de combustível em momentos diferentes. Um órgão regulador realiza uma análise de segurança, descobrindo que o componente responsável pela contagem dos pulsos usados para medir o combustível está adulterado. Observe que este tipo de modificação pode ser parcial, quando um componente parasita é adicionado, ou total, quando um componente totalmente diferente é realocado. Contudo, note que o ataque não interfere em qualquer proteção criptográfica existente na memória, no processador e no *software*. Este tipo de ataque aconteceu no Brasil, sendo detectado em diferentes oportunidades pelo Inmetro (LEITÃO; VASCONCELLOS; BRANDÃO, 2014). Diante do exposto, é necessário melhorar a segurança das medições nos MI, uma vez que mesmo com altos padrões de segurança, existem componentes vulneráveis que fazem parte da cadeira LR.

2.2 VULNERABILIDADES NOS DISPOSITIVOS INTELIGENTES

Os dispositivos inteligentes precisam aumentar a confiabilidade devido ao grande número de aplicativos em que estão envolvidos e as informações geradas por eles (CHHETRI et al., 2017; Melo Jr et al., 2019). Uma característica importante desses dispositivos são as limitações de processamento, memória e *software* (CAI; ZUHAIRI, 2017; LIAO et al., 2020). Essas características influenciam diretamente na segurança, principalmente no que diz respeito a integridade, devido que, sua implementação está diretamente relacionada aos recursos de computação do dispositivo (*software* e *hardware*) (XU; WENDT; POTKONJAK, 2014; DABBAGH; RAYES, 2019). A localização do dispositivo inteligente também é relevante, especialmente quando faz parte de em ambientes considerados hostis. Para esta tese, um ambiente hostil é um local onde entidades maliciosas podem ter acesso físico ao dispositivo.

Nesta subseção, propomos uma classificação das vulnerabilidades encontradas em dispositivos inteligentes, especificamente em medidores inteligentes. A classificação é orientada do ponto de vista de quem deseja descobrir/identificar vulnerabilidades de um IM. Este pode ser tanto um atacante, quanto uma entidade metrológica que avalia a segurança do dispositivo. A classificação de vulnerabilidades baseia-se em dois aspectos principais: ambiente operacional e nível de acesso (ver Figura 1).

Na Figura 1 temos duas categorias: Ambiente e Acessos. Ambiente refere-se a um espaço (com pessoal e ferramentas) para analisar um ou mais componentes do

MI. Os ambientes propostos são: análise de *software*, equipamentos de *hardware* simples, laboratório de *hardware* especializado e fabricação. Com base neles, as vulnerabilidades são agrupadas. Observe que é possível encontrar a mesma vulnerabilidade em dois ambientes diferentes. Isso se deve às características do próprio ambiente, no que diz respeito à qualificação do pessoal e das ferramentas usadas para a busca das vulnerabilidades. Nesse contexto, cada ambiente também possui seu próprio nível de complexidade. Essa complexidade é representada horizontalmente, em ordem crescente, da esquerda para a direita, onde as vulnerabilidades localizadas à esquerda são menos complexas do que as da direita.

Os Acessos representam as camadas do MI que devem ser acessadas em busca das vulnerabilidades. Propusemos 3 acessos: Acesso Interface Homem-Máquina, Acesso Interface Homem-Máquina e Rede, e Acesso aos componentes internos. Cada vulnerabilidade está associada a um tipo de acesso que, por sua vez, está relacionado ao nível de complexidade da vulnerabilidade. Portanto, acessar as Interfaces Homem-Máquina é mais fácil do que acessar os componentes internos do MI. A seguir explicamos cada um dos ambientes e vulnerabilidades.

2.2.1 Ambiente

Os ambientes são espaços projetados para analisar os MI. Para cada ambiente, diferentes partes do MI são analisadas, incluindo o *software* e o *hardware*. Nesse sentido, são necessários diferentes conhecimentos para cada ambiente, incluindo as ferramentas necessárias para encontrar vulnerabilidades. A seguir, cada um dos ambientes é explicado.

Análise de Software: Neste ambiente, testes funcionais são aplicados ao *software* em busca de possíveis falhas, que na pior das hipóteses podem levar a *buffer overflow* (DAVIDSON et al., 2016; MULLEN; MEANY, 2019). Adicionalmente, são analisadas as iterações do usuário através das Interface Homem-Maquina (IHM), especialmente, durante validação de dados (MITROPOULOS et al., 2016; SARRAB; ALNAELI, 2019).

Equipamentos *hardware* simples: Nesse ambiente, o MI é explorado fisicamente. Ferramentas de análise de *hardware* com custo médio a médio-alto são usadas para este ambiente. Essa característica limita os tipos de testes físicos realizados no MI, incluindo o conhecimento exigido pela pessoa para realizá-los. Pessoas neste ambiente sabem usar as ferramentas e identificam vulnerabilidades, mas durante o processo podem danificar o MI. Entre as ferramentas utilizadas neste ambiente estão analisadores de sinais lógicos e digitais, analisadores de barramentos, microscópios, entre outras ferramentas que permitam realizar análise do *hardware*.

Laboratório de *hardware* especializado: Este tipo de ambiente é avançado e possui equipamentos ainda mais especializados para análise de *hardware*. Este ambiente é encontrado em laboratórios de desenvolvimento e pesquisa de *hardware*. Alguns dos equipamentos são: analisadores digitais e lógicos de alta frequência, analisadores de raios X 2D e 3D para PCBs, analisadores de memória, analisadores de chip, entre outros. Este ambiente é formado por especialistas em suas áreas de atuação, com capacidade de identificar vulnerabilidades, causando efeitos mínimos no *hardware*.

Fabricação: Durante a produção de *software* e *hardware*, é normal terceirizar o desenvolvimento e estabelecer cadeias de produção localizadas em diferentes países (KONSTANTINOU; KELIRIS; MANIATAKOS, 2016). No entanto, essas cadeias de produção podem trazer problemas de segurança que envolvem a integridade e privacidade do dispositivo inteligente (KONSTANTINOU; KELIRIS; MANIATAKOS, 2016). No caso do *software*, funcionalidades maliciosas podem ser adicionadas pelos desenvolvedores (CHHETRI et al., 2018). No caso de *hardware*, subcontratados podem roubar o design para fazer cópias baratas de um produto legítimo (HASSIJA et al., 2020). Também é possível adicionar componentes não projetados para fins maliciosos, como roubo de informação, espionagem, entre outros (ZHANG et al., 2018).

2.2.2 Acessos

Acesso agrupa vulnerabilidades em três grandes categorias: Acesso Interface Homem-Máquina (IHM), Acesso Interface Homem-Máquina e Rede (IHMR) e Acesso aos componentes internos. Elas possuem um ordem, onde as localizadas mais à esquerda requerem conhecimentos superficiais e menos invasivos ao MI e seus componentes (*software* e *hardware*). Por sua parte, as localizadas mais a direita são vulnerabilidades que precisam de conhecimentos avançados tanto em *software* quanto em *hardware*. A seguir, cada um deles é explicado:

Acesso Interface Homem-Máquina: nesse tipo de acesso, o invasor tem pouco ou nenhum conhecimento sobre *software* ou *hardware* do MI. O principal objetivo é explorar e identificar as interfaces do *software* e portas de configuração acessíveis. Por exemplo, interfaces de usuários no *software* e portas físicas como as seriais, *Universal Serial Bus* (USB), *thunderbolt*, entre outras.

Acesso Interface Homem-Máquina e rede: nesta categoria, o invasor é capaz de analisar informações sobre o tipo de rede de comunicação interna (como barramento) e entre dispositivos (como WiFi), incluindo também a identificação dos protocolos utilizados nas comunicações e os componentes envolvidos. Baseado nesse tipo de informações, é possível encontrar vulnerabilidades. Entre os tipos de redes de

comunicação incluídas estão as internas, como o barramento, *Joint Test Action Group* (JTAG), e pinos de configuração; e também as redes de comunicação externas, como Wifi, *Ethernet*, e *Bluetooth*.

Acesso aos componentes internos: o invasor realiza uma análise abrangente para entender o comportamento do *software*, *hardware* e interações físicas das medições do MI. Nesse cenário, é possível alterar ou modificar o comportamento do *software* e *hardware* de forma controlada e consciente.

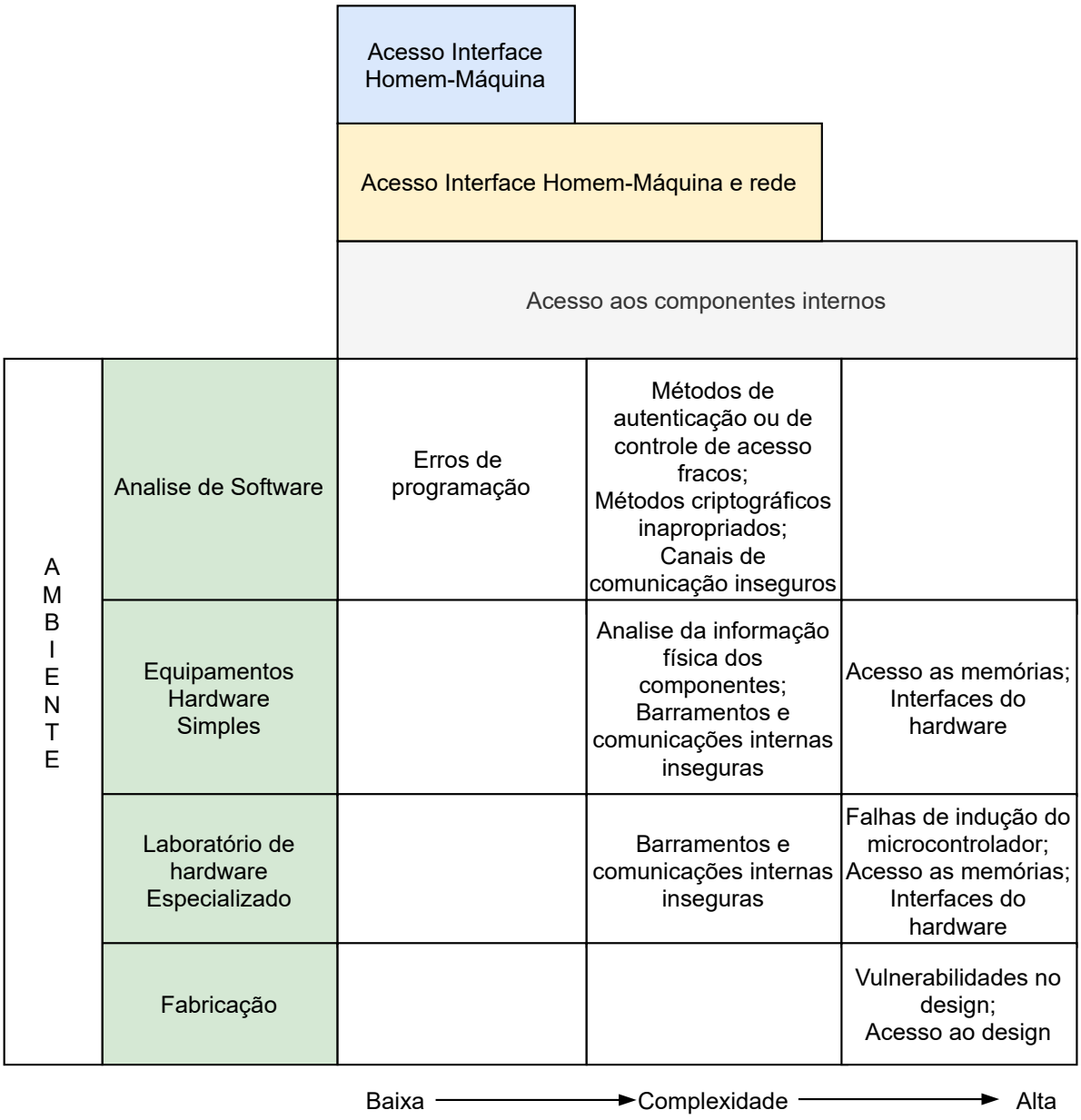

Figura 1 – Taxonomia vulnerabilidades nos dispositivos inteligentes

2.2.3 Vulnerabilidades do *Software*

Diversas vulnerabilidades em *software* nos dispositivos inteligentes têm sido estudadas na literatura (KONSTANTINOU; KELIRIS; MANIATAKOS, 2016; CHEN et al., 2018a; AHMADVAND; PRETSCHNER; KELBERT, 2019). Para a classificação das vulnerabilidades, também foram consideradas aquelas mencionadas pelo *Open Web Application Security Project* (OWASP) [1], uma comunidade de renome mundial que fornece ferramentas e documentos para melhorar a segurança de *software*. Embora esta tese se concentre no *hardware*, consideramos importante apontar vulnerabilidades que envolvem a parte de *software* em dispositivos como medidores inteligentes e dispositivos no paradigma do IoT. A seleção das vulnerabilidades é baseada na seguinte premissa, ter acesso as interfaces de usuário do *software*.

Erros de programação: Erros de programação podem levar a várias vulnerabilidades no *software*, como por exemplo *buffer overflow*, onde problemas de manipulação de memória permitem que partes legítimas do *software* sejam sobrescritas (CAI; ZUHAIRI, 2017; MULLEN; MEANY, 2019). Também é possível expor dados sensíveis que devem ser mantidos em segurança, como senhas, variáveis de medição e chaves criptográficas (AHMADVAND; PRETSCHNER; KELBERT, 2019; SARRAB; ALNAELI, 2019; ASGHAR et al., 2017). Existe a possibilidade de se ter código seguro, mas utilizando bibliotecas que possuem erros de programação, comprometendo indiretamente o software.

Métodos de autenticação e controle de acesso fracos: As funções de autenticação e gerenciamento de sessão do *software* devem ser implementadas adequadamente (WANG et al., 2016a; SARRAB; ALNAELI, 2019). O uso de nome de usuário e senha é o principal método para realizar o controle de acesso. No entanto, para ser considerado seguro, é necessário adotar alguns recursos de segurança nas credenciais de acesso, como tamanho da senha, uso de caracteres especiais, entre outros. Estas devem ser conhecidas apenas pelo usuário e não devem ser compartilhadas com terceiros. Se as credenciais e os métodos de controle de acesso forem fracos, terceiros podem suplantar um usuário legítimo. O acesso se torna mais perigoso quando as contas afetadas têm privilégios de administrador. Desta forma, é possível encontrar implementações fracas, implementadas erroneamente e até mesmo com erros de configuração.

Métodos criptográficos inapropriados: A criptografia é um método usado para manter a confidencialidade, integridade e autenticidade dos dados nos dispositivos. No entanto, sua implementação pode incluir vulnerabilidades, como o uso indevido de aplicações criptográficas (KOTESHWARA; DAS, 2017), armazenamento incorreto de chaves (LI et al., 2018), algoritmos com defeitos conhecidos (GUPTA; GUPTA; SINGH,

[1] link: https://owasp.org/www-community/vulnerabilities/

2019), implementação de criptografia proprietária (MEIJER; MOONSAMY; WETZELS, 2020), entre outros (SALLAM; BEHESHTI, 2019).

Canais de comunicação inseguros: A comunicação é essencial em dispositivos inteligentes. Entretanto, as limitações de memória e processamento podem dificultar a implementação de métodos de comunicação adequados (TOMIĆ; MCCANN, 2017). Dada esta limitação, os projetistas devem analisar e escolher métodos de comunicação em equilíbrio com os componentes, protocolos e recursos de comunicação suportados pelo dispositivo (ALLADI et al., 2020). Erros neste sentido podem levar à escolha de protocolos inseguros como o TELNET (PROKOFIEV; SMIRNOVA; SILNOV, 2017), SNMP (WILLIAMS et al., 2017), e configurações inadequadas de IPV4 e IPV6 (GHALEB et al., 2016). Há também a possibilidade de utilizar bibliotecas encarregadas de realizar os processos de comunicação, que incluem vulnerabilidades conhecidas, expondo as informações transmitidas (SHU; YAO; BERTINO, 2015).

2.2.4 Vulnerabilidades no *Hardware*

Existe uma ampla gama de ataques em dispositivos inteligentes que podem ser realizados através de vulnerabilidades no *hardware* (LI; ATTARMOGHADDAM, 2018; VALEA et al., 2019). Nesta seção vamos explicar as principais vulnerabilidades para sistemas inteligentes com base nos estudos desenvolvidos por (LI; ATTARMOGHADDAM, 2018; PRINETTO; ROASCIO, 2020; GIECHASKIEL; RASMUSSEN, 2019; XUE et al., 2020; VALEA et al., 2019).

Interfaces e portas do *hardware*: Desenvolvedores utilizam este tipo de interfaces e portas para encontrar e eliminar erros de programação (*debugging*), testar componentes do dispositivo, verificar e diagnosticar interconexões dos circuitos, entre outras coisas (LI; ATTARMOGHADDAM, 2018; KONSTANTINOU; MANIATAKOS, 2019). De fato, estas características são úteis no processo de desenvolvimento, mas trazem problemas de segurança como inspeção de dados, injeção de processos e código, e manipulação das interrupções no sistema (Tanjidur Rahman et al., 2018). Por exemplo, através da interface JTAG (Joint Test Action Group) é possível recuperar o binário do *software* ou *firmware* carregado pelo dispositivo. A interpretação do binário, pode ser realizado com ferramentas como o GBD [2] e o IDA Pro [3], que transformam a informação coletada em linguagem de máquina (KONSTANTINOU; MANIATAKOS, 2019).

Acesso as Memórias: As memórias de um MI podem armazenar parâmetros de calibração, chaves criptográficas e até mesmo partes do *firmware* (YU et al., 2017).

[2] https://www.gnu.org/software/gdb/download/
[3] https://hex-rays.com/ida-pro/

Pessoas com acesso físico aos pinos de memória e interfaces de comunicação podem ter acesso a esses dados e copiar o conteúdo. Esse processo é chamado de *memory dump* (Tanjidur Rahman et al., 2018). O ideal é manter os dados armazenados com segurança, por exemplo criptografados. No entanto, esse processo pode ter um custo proibitivo em alguns dispositivos inteligentes que possuem recursos limitados de memória e CPU.

Análise da informação física dos componentes: As informações disponíveis sobre aspectos físicos dos componentes (como por exemplo nomes, números de série e informações de pinagem) são úteis para entender seu funcionamento e identificar vulnerabilidades. Com essas informações também é possível estabelecer pinos usados para comunicação com outros componentes (que por sua vez podem ter suas próprias vulnerabilidades exploráveis), ou até mesmo trocar um componente por outro modificado (Hallmans et al., 2015; ZHANG et al., 2018; SHWARTZ et al., 2017)

Barramentos e comunicações internas inseguros: Os componentes internos de um dispositivo devem se comunicar e interagir uns com os outros. No *hardware*, isso é feito por meio de canais físicos chamados barramentos (*bus*) e protocolos de comunicação serial, como o *Inter-Integrated Circuit* (I2C) e *Serial Peripheral Interface* (SPI). Esses protocolos usam o modelo *Master/Slave*, onde o *Master* é encarregado de iniciar e terminar as comunicações. Nesse tipo de modelo, é comum que os componentes compartilhem o mesmo barramento, de forma que as mensagens transferidas também são "escutadas" pelos demais componentes (LIU et al., 2019; AHMED; MATHUR; OCHOA, 2020). Em consequência, é possível acoplar dispositivos externos no barramento e capturar mensagens trafegadas (PALEY; HOQUE; BHUNIA, 2016; KHELIF et al., 2020). No pior dos casos, um atacante pode forçar a inserção de mensagens maliciosas através do barramento (IEHIRA; INOUE; ISHIDA, 2018)

Falhas de indução do microcontrolador: Flutuações na tensão e no *clock* do microcontrolador podem causar comportamentos errôneos e inesperados (KORAK; HOEFLER, 2014). No caso da tensão, as flutuações podem revelar informações sobre a implementação de aplicações criptográficas e informações confidenciais como chaves privadas (MALDINI et al., 2019). Todo microcontrolador precisa de um *clock* para realizar diferentes ações e operações, como controlar a velocidade de execução das instruções, a velocidade de transmissão dos sinais de comunicação, o tempo para realizar as conversões analógicas para digitais, entre outras coisas (REVERTER; GASULLA, 2021). Quando o *clock* é externo, ou seja, fora do microcontrolador, este fica vulnerável à manipulação por terceiros (MICHAEL, 2014). Consequentemente, um atacante pode alterar o comportamento do microcontrolador. Em (KAZEMI et al., 2020), os autores demostraram que uma manipulação controlada do *clock* evita a execução de instruções

específicas pelo microcontrolador.

Vulnerabilidades no projeto: Uma vulnerabilidade pode ser inserida inadvertidamente durante os estágios de projeto e produção de *hardware*. De acordo com (PRINETTO; ROASCIO, 2020), dois fatores principais são a causa: erros e falhas. Um erro é uma inconsistência entre a especificação do projeto e a implementação final. As falhas, por sua vez, estão presentes em funções desenvolvidas e testadas pelos projetistas. No entanto, os testes podem não cobrir ações potencialmente perigosas, seja por omissão ou desconhecimento da vulnerabilidade. Exemplos recentes de vulnerabilidades baseadas em falha são o *Meltdown* (LIPP et al., 2020) e *Spectre* (KOCHER et al., 2020). Essas duas vulnerabilidades devem-se a uma otimização nos processadores, permitindo que os aplicativos acessem informações confidenciais carregadas em memória.

Acesso privilegiado a informações de projeto: Linhas de produção são estabelecidas no mundo para a produção de *hardware* e seus componentes (HU et al., 2016). Essas linhas de produção geram grande preocupação por parte dos projetistas, pelo fato de terem que entregar Propriedade Intelectual (PI) para empresas terceirizadas. Dois principais problemas podem ocorrer nas linhas de produção: (i) roubo da PI, ou seja, empresas terceirizadas podem copiar o projeto e distribuí-lo como sendo de sua propriedade (HU et al., 2016); e (ii) manipulação de projeto, *chips*, processadores, PCBs e outros componentes para incluir funções adicionais por motivos maliciosos, como monitoramento e espionagem (TEHRANIPOOR; KOUSHANFAR, 2010; HU et al., 2016). Este processo também é chamado de *Hardware Trojan* (HT) (Tanjidur Rahman et al., 2018; XUE et al., 2020).

2.3 IDENTIFICAÇÃO E AUTENTICAÇÃO COMO MÉTODO DE SEGURANÇA

A identificação em Ciência da Computação é a capacidade de reconhecer exclusivamente um usuário ou entidade que faz parte de um sistema ou aplicativo (SANDHU et al., 2012). A autenticação é a forma como o sistema ou aplicativo verifica e prova que o usuário é quem afirma ser. Como premissa, o identificador de um usuário (ou sua identidade) deve ser algo único para o sistema ou aplicação, como por exemplo um *Personal Identification Number* (PIN), endereço IP, chaves criptográficas, digital, entre outras. Como premissa para a autenticação, as informações fornecidas pelo usuário devem ser de seu conhecimento exclusivo e não devem ser compartilhadas com terceiros.

O uso combinado da identificação e autenticação constitui o método de controle de acesso mais comum para sistemas, e pode ser desenvolvido tanto em *software* como em *hardware*. As implementações mais comuns são baseadas em *software*, onde

o usuário cria uma identificação e senha no sistema, e o próprio sistema a autentica. Soluções baseadas em *hardware* são menos populares e têm um custo maior. Atualmente, *tokens* e *dongles* USB (FORTE; BHUNIA; TEHRANIPOOR, 2017) são exemplos de implementações comerciais.

Atualmente existem dispositivos capazes de realizar operações de identificação e autenticação de forma autônoma, sem interatividade humana. Esta característica pode ser encontrada em alguns dispositivos inteligentes, como IoT. A identificação e autenticação podem ser realizadas por meio de gerenciadores de inicialização como um *bootloader*, antes que o *software* seja carregado na memória. Outras soluções mais avançadas incluem o uso de componentes de *hardware* e *software* num mesmo dispositivo, como uma solução mais robusta (LEE; MARKANTONAKIS; AKRAM, 2016). Componentes como o TPM (*Trusted Platform Module*) permitem gerenciar o uso e armazenamento de chaves e *hashes*, por exemplo (BRASSER et al., 2018).

2.3.1 Identidades, identificadores, e segurança

A ideia de geração de identidades não é nova e pode ser vista como um complemento aos processos de identificação e autenticação. Dentro da Computação, podemos entender as identidades como características ou atributos que podem ser usados individualmente ou em conjunto, permitindo verificar, identificar e autenticar uma pessoa ou dispositivo (HANSEN; SCHWARTZ; COOPER, 2008). As definições de identidades e identificadores, em alguns casos, podem ser confundidas. Para esta tese, definimos identificadores como valores que podem ser redefinidos, e identidades como imutáveis e duradouras no tempo, similar ao *fingerprint*.

Na era dos dispositivos inteligentes, os dispositivos IoT e os sistemas ciber-físicos enfrentam novos desafios, como garantir a segurança dos componentes internos e das informações processadas e transferidas por eles (LI; SONG; ZENG, 2018). Isso torna-se mais relevante dado que esses dispositivos podem possuir sensores que interagem com o ambiente e auxiliam na tomada de decisões (WANG et al., 2016b). O uso de identidades baseadas em dispositivos mostra-se promissor como forma de enfrentar os desafios, detectando mudanças nos componentes que compõem a identidade. Portanto, é necessário explorar formas de aprimorar as identidades junto com os métodos de segurança atuais.

Uma área também promissora é o uso de características físicas e contexto físico dos dispositivos (MELO; MACHADO; CARMO, 2018; IVANOV; WEIMER; LEE, 2018). Basicamente, as características físicas podem ser entendidas como características internas do dispositivo, e o contexto físico como características físicas que pertencem ao

ambiente no qual o dispositivo está. Nas próximas subseções exploramos e explicamos esses conceitos em profundidade.

2.4 CARACTERÍSTICAS FÍSICAS E CONTEXTO FÍSICO

Esta subseção abordará os conceitos de características físicas e contexto físico de um dispositivo, sua aplicabilidade nos cenários de computação atuais e seus benefícios. Também explicamos como esses conceitos são aplicáveis à segurança da informação.

2.4.1 Características físicas

Na área de Computação, as características físicas de uma entidade vêm sendo abordadas há alguns anos, principalmente para os processos de identificação e autenticação. Um exemplo é o caso da biometria. Segundo (LOZOYA-SANTOS et al., 2019), a biometria explora características físicas e comportamentais com o intuito de reconhecer pessoas através do rosto, impressão digital, voz, íris, entre outras. Entre as técnicas para modelar a biometria na computação estão métodos de dados teóricos, modelos matemáticos, métodos analíticos e técnicas de simulação computacional (YAGER; DUNSTONE, 2010). Com isso, é possível tomar uma característica física de um ser humano e transformá-la em um valor digital que representa o ser humano de forma única e inequívoca.

Recentemente, pesquisadores aplicaram o conceito de características físicas para identificar *hardware* (PAPPU et al., 2002; KUMAR; DHANUSKODI; KUNDU, 2014). Essa técnica é chamada de *Physical Unclonable Function* (PUF). Semelhante à biometria, os PUFs usam as características físicas intrínsecas do *hardware* para obter valores únicos (VIJAYAKUMAR; PATIL; KUNDU, 2016). Esses valores de PUF são obtidos de variações físicas durante o processo de fabricação do *hardware*. Entre os *hardwares* comumente usados para PUF estão *chips*, memórias e *lasers* (SUTAR; RAHA; RAGHUNATHAN, 2016; JELOKA et al., 2017; HERDER et al., 2014). Os valores de um PUF são validados por modelos estatísticos e matemáticos onde é possível identificar a viabilidade do seu uso. Na seção 2.4.5, explicamos com mais profundidade o conceito de PUF, suas características, aplicações e formas de validação.

2.4.2 Contexto Físico

O contexto físico pode ser definido a partir de eventos físicos de um sistema capturados por sensores em um processo de interação ciber física (HABIB; LEISTER, 2015; MELO; MACHADO; CARMO, 2018). Os eventos capturados descrevem o contexto do sistema e podem influenciar estratégias de tomada de decisão e controle. Dessa forma,

o contexto físico pode ser entendido também como informação que representa o estado corrente do ambiente do sistema.

A captura de informações pode ser feita de duas maneiras: direta ou indireta. (IVANOV; WEIMER; LEE, 2018). No caso direto, as informações são coletadas sem a necessidade de qualquer pré-processamento antes de serem utilizadas pelo dispositivo, por exemplo, a temperatura ambiente medida por um sensor. No caso dos indiretos, as informações coletadas devem ser processadas antes de serem utilizadas, por exemplo, um sistema que analisa amostras de sangue.

O contexto físico pode ser usado por diferentes dispositivos e sistemas para auxiliar na tomada de decisão, como é o caso dos *Cyber Physical Systems* (CPS). De acordo com o *National Institute of Standards and Technology* (NIST), CPS são sistemas que interagem com componentes do mundo digital, analógico e físico (Griffor Edward R., Greer Christopher, Wollman David A., 2017). Sistemas chamados CPS são encontrados em *Smart Grids*, *Smart Cities*, *Smart Factories*, *Smart Buildings*, e *SmartHomes* (SHI et al., 2011). Sistemas relacionados à quarta revolução industrial, ou a Indústria 4.0, também se valem do contexto físico para realizar tarefas inteligentes, combinando procedimentos e processos com objetos, sensores e atuadores em tempo real (SHI et al., 2011; CAMARINHA-MATOS et al., 2015; GUIZANI, 2019). O contexto físico passa a ser um eixo vital para a tomada de decisão dentro do CPS e Industria 4.0, uma vez que a interação humana é cada vez mais escassa. A tomada de decisão depende dos processos e ações de atuadores, sensores e dispositivos inteligentes com seu contexto físico. Em contrapartida, erros de atuadores, sensores e dispositivos inteligentes ao capturar ou compreender seu contexto físico podem conduzir a problemas na execução de suas ações e até mesmo a falhas de segurança (CHHETRI et al., 2017)

2.4.3 Aspectos e características necessárias de um contexto físico.

Uma premissa para considerar um contexto físico é ser derivado de algum fenômeno físico descrito por grandezas físicas (MELO, 2018). Velocidade, temperatura e massa são apenas alguns exemplos. Estas grandezas físicas devem ser próprias do ambiente onde o dispositivo se encontra, ou produzidas pelo sistema e demais dispositivos associados a ele. A seguir apresentamos as considerações descritas por (MELO, 2018) sobre aspectos relevantes em contextos físicos.

- Sistemas físicos são influenciados por fenômenos físicos próprios do ambiente, os quais podem ser entradas, estados ou saídas do sistema;
- Os eventos físicos devem ter uma associação intrínseca ao estudo do sistema;

- Coalocação descreve a condição onde dois ou mais dispositivos encontram-se em uma mesma localização física, dentro do escopo de observação delimitado por seu contexto físico;
- Simultaneidade descreve a condição onde dois ou mais dispositivos podem capturar o mesmo contexto físico, ao mesmo tempo.

Basicamente, estas considerações indicam a correlação que deve ter um contexto físico e o sistema. Os contextos físicos também podem ser classificados de acordo com suas propriedades. Em (MELO, 2018) o autor os classifica da seguinte forma:

- **Causalidade:** os eventos classificados neste tipo de contexto tem dois estados, estimulado e espontâneo. No estimulado o evento é derivado de ações realizadas por entidades sobre o sistema. No caso do espontâneo, o evento físico é resultado de ações próprias do sistema sem requer ações por parte de entidades;
- **Previsibilidade:** o evento pode ser previsível ou não previsível. Eventos previsíveis são aqueles onde modelos matemáticos podem prever quando o contexto físico acontecerá, por exemplo aplicar *Machine Learning*. Por sua vez, os imprevisíveis são aqueles em que a ocorrência do evento físico não é determinística;
- **Unicidade:** os eventos podem ser únicos ou recorrentes. Único refere-se a eventos em sistemas que são destruídos ou inutilizados após o evento. No caso dos recorrentes, refere-se àqueles eventos que se repetem indefinidamente. Um exemplo são mecanismos encontrados em relógios;
- **Descrição quantitativa:** eventos podem ser monovariável, multivariável dependente ou multivariável independente. No caso de variáveis individuais, são eventos descritos por uma única grandeza física, como a aceleração de um veículo. Por sua vez, multivariável refere-se a um evento físico que é descrito por mais de uma grandeza física, e correlacionadas entre elas. por exemplo, evento físico que é descrito por um acelerômetro e uma célula de carga. Por último, no caso de multivariável independente são eventos que descrevem diversas grandezas físicas, mas que não existe nenhuma correlação entre elas.

2.4.4 Características físicas e contexto físico aplicados a segurança da informação

O contexto físico e as características físicas têm uma propriedade comum de interesse para a área de segurança da informação: a unicidade. Para esta tese, definimos unicidade como as propriedades de um objeto que o tornam único. Algumas formas para

a obtenção da unicidade são através das características físicas do ambiente captadas por sensores (KARAPANOS et al., 2015) ou através de características próprias do *hardware* (ADAMES; DAS; BHANJA, 2016). A unicidade torna-se relevante para a segurança da informação quando pode ser vinculada a métodos e técnicas existentes, por exemplo para gerar *True Random Number Generators* (TRNGs) (MARTIN et al., 2016). Além disso, o fato de o contexto físico ter uma origem baseada no mundo físico ou em *hardware*, oferece maior robustez em relação às soluções baseadas em *software* (FOURNARIS; LAMPROPOULOS; KOUFOPAVLOU, 2017; DEMIGHA; LARGUET, 2021).

Assim as características do *hardware* e do ambiente ao qual um dispositivo pertence pode ser explorado como parte dos métodos e técnicas de segurança (MOIS; SANISLAV; FOLEA, 2016). Essencialmente, soluções que exploram o uso de componentes integrados no dispositivo, como sensores, memórias e processadores. Uma característica deste tipo de soluções é a capacidade de implementá-las sem exigir altos recursos computacionais, sendo útil em dispositivos inteligentes que possuem restrições de armazenamento, processamento e consumo de energia.

Atualmente é possível encontrar dispositivos inteligentes com sensores de temperatura, frequência cardíaca, giroscópios, entre outros. Esses sensores podem atuar como parte de técnicas ou métodos de proteção com base em características físicas e/ou contexto físico. Isso abre novas oportunidades para explorar a segurança da informação, favorecendo o desenvolvimento de soluções que combinem segurança de *software* e *hardware*.

2.4.5 *Physical Unclonable Function*: definição e aplicabilidades

Physical Unclonable Function (PUF) são valores únicos e imprevisíveis gerados através das caraterísticas físicas intrínsecas obtidas durante o processo de fabricação do *hardware* (GUAJARDO et al., 2007). A unicidade e imprevisibilidade favoreceu o uso de PUF dentro da área da segurança da informação (WACHSMANN; SADEGHI, 2014). O processo para obter os valores PUFs é realizado através de desafios e respostas, onde cada desafio possui uma única resposta. Formalmente, o PUF pode ser representado com a seguinte função:

$$f_p(D_i) = R_i \tag{2.1}$$

Onde f_p representa a função PUF, D_i é o desafio e R_i a resposta para o desafio. Para ser considerado PUF, algumas propriedades devem existir (HERDER et al., 2014; MELO, 2018), mais especificamente:

- **Fácil de verificar:** Quando f_p usa um desafio D_i, a resposta R_i é entregue em intervalo de tempo curto;
- **Difícil de estimar:** Assume-se que respostas R_i e R_j provenientes de desafio D_i e D_j (sendo $D_i \neq D_j$), aplicado em f_p não fornecem informação relevante uma da outra. Inclusive, PUFs fisicamente iguais $f_p()$ e $f'_p()$ que usam um mesmo desafio D_i entregam respostas totalmente diferentes ($R_i \neq R'_i$).
- **Difícil de prever:** Dado que o atacante tenha o desafio D_i e a resposta R_i, ele não deve poder deduzir a resposta R_j se tem o desafio D_j;
- **Inviolável:** Assume-se que o PUF é construído de modo que evidencie qualquer tentativa física de adulteração. Isto implica que ante ataques físicos, a estrutura do PUF deve ficar danificada ao ponto de ser inutilizado ou entregar informações errôneas.

Os PUFs podem ser classificados em dois grandes grupos: *Weak* PUFs e *Strong* PUFs. Esta classificação depende da quantidade de desafios e respostas do PUF. Nesse sentido, os PUFs *Weak* são aqueles que possuem conjunto limitado de desafios e respostas, enquanto os *Strong* PUFs são aqueles que possuem uma variedade ampla de desafios e respostas. Diferentes técnicas são usadas para desenvolver PUFs, contudo, predominam aquelas baseadas em chips e memórias. Os PUFs baseados em memória são considerados *Weak*, enquanto os PUFs baseados em chip podem ser classificados como *Weak* ou *Strong*. A força dos PUFs baseados em chips dependerá da forma como foi desenvolvido.

2.4.5.1 Avaliação dos PUFs

Para considerar as respostas do PUF como válidas e seguras, devem primeiro ser analisadas. Um método amplamente adotado é o uso da *Hamming Distance* (HD). HD é uma medida de similaridade, que permite determinar a correspondência ou diferença entre dois *string* de *bits* do mesmo tamanho. Por meio da HD, é possível identificar a viabilidade das respostas de um PUF, representada por dois critérios: Inter-HD e Intra-HD (GUAJARDO et al., 2007; HERDER et al., 2014).

Intra-HD: representa a distância entre as respostas de um mesmo PUF usando o mesmo desafio. O valor Intra-HD ideal para um PUF é 0; no entanto, devido a vários fatores, como variações de temperatura e fonte de alimentação, esse valor é diferente de 0.

Inter-HD: representa a unicidade das respostas entre PUF fisicamente iguais aplicando o mesmo desafio. Espera-se que haja uma variação significativa entre as

respostas, o que permite corroborar a independência entre PUF. Para ser considerada uma resposta válida, o valor ideal do Inter-HD deve ser próximo a 50%.

Em resumo, podemos dizer que intra-HD é útil para conhecer a estabilidade da resposta ao longo do tempo para um PUF específico. Por sua vez, o Inter-HD permite conhecer a diferença de respostas entre PUFs que são fisicamente iguais.

2.4.5.2 *Weak* PUFs

Os *weaks* PUFs são baseados no comportamento das memórias que existem num dispositivo. Entre os tipos de memória para os dispositivos, encontra-se as voláteis e não voláteis. Uma característica das memórias voláteis é que, quando são inicializadas, o preenchimento dos *bits* de informação é feito com valores aleatórios. As técnicas de PUF baseadas em memória exploram esse processo de preenchimento das memórias voláteis, visto que é causado por características físicas obtidas durante o processo de fabricação. Essas características fazem com que os valores obtidos da memória sejam únicos, inclusive se são fisicamente iguais, por exemplo, mesmo modelo (JELOKA et al., 2017).

Conforme indicado na seção 2.4.5, os PUFs trabalham com o modelo de desafios e respostas. No caso de PUFs baseados em memória, o desafio é a energia usada para inicializar a memória e a resposta são os valores com os quais a memória é preenchida. Nesse sentido, existe uma correlação direta entre os valores que preencheram a memória e a voltagem utilizada para a inicialização. Entre as memórias voláteis usadas pelos PUFs estão as *Static Random Access Memory* (SRAM) (SHIN et al., 2016) e *Dynamic Random Access Memory* (DRAM) (KURINEC; INIEWSKI, 2013) .

A Figura 2 mostra o diagrama de uma célula de memória SRAM apresentada por (ZHANG; WANG; ZHANG, 2013). A SRAM está composta por 4 transistores de carga (N1, N2, P1, P2) e dois transistores de acesso (N3 e N4), conectados ao barramento BL e BLC para ler e escrever. Os transistores compartilham as linhas *Voltage Commom Collector* (VCC) e terra, permitindo que durante a inicialização da memória eles recebam a mesma quantidade de VCC. A inicialização induz a cada célula da memória tomar o valor de 0 ou 1.

As memórias DRAM são consideradas como uma alternativa as amplamente conhecidas SRAM. Similar as SRAM, as DRAM são voláteis e perdem os dados armazenadas quando são desenergizadas. As células DRAM consistem em um capacitor que armazena carga acima ou abaixo de um determinado limite, indicando um valor lógico, e um transistor de *gatekeeper* que controla o acesso ao capacitor de armazenamento. A

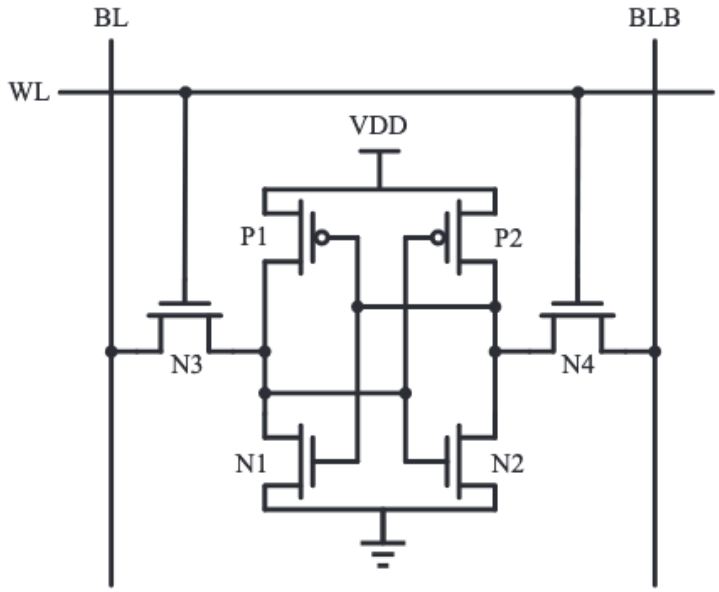

Figura 2 – Célula do SRAM PUF

forma como funciona uma DRAM é através de matrizes de memória, onde os acessos são realizados através das linhas que armazenam os conjuntos de bits. Na Figura 3 apresentamos um modelo da arquitetura de um DRAMs apresentada por (TEHRANIPOOR et al., 2017b). Uma desvantagem das DRAMs em relação às SRAMs é a necessidade de atualizar constantemente o conteúdo armazenado na célula. Isto deve-se a um constante esmorecimento inerente do design das memórias DRAM.

A segurança em *weak* PUFs está em impedir o acesso de terceiros ao conteúdo da memória. Se as respostas de PUF puderem ser obtidas por meio de ataques invasivos, a segurança será comprometida (RÜHRMAIR; HOLCOMB, 2014). Nesse sentido, os autores em (NEDOSPASOV et al., 2013) demonstraram a possibilidade de utilizar um *laser* para obter mapas de memória. O ataque é semi-invasivo e não destrói a SRAM utilizada, permitindo que o dispositivo continue em operação sem afetar as características físicas internas utilizados pelo PUF. Um ponto negativo para este tipo de ataque é não conseguir desenvolver um modelo preditivo que permita conhecer as respostas esperadas de PUFs fisicamente iguais. Em outras palavras, o procedimento é individual e deve ser realizado para cada PUF.

Outros tipos de ataques em *weak* PUFs são aqueles relacionados a clonagem de memória (HELFMEIER et al., 2013). As interfaces de acesso à memória e os barramentos de dispositivos são usados para enviar e receber respostas do PUF. Os invasores que têm acesso a essas interfaces podem conhecer o conteúdo armazenado na memória (HELFMEIER et al., 2013). Uma característica geralmente encontrada nos *weaks* PUFs

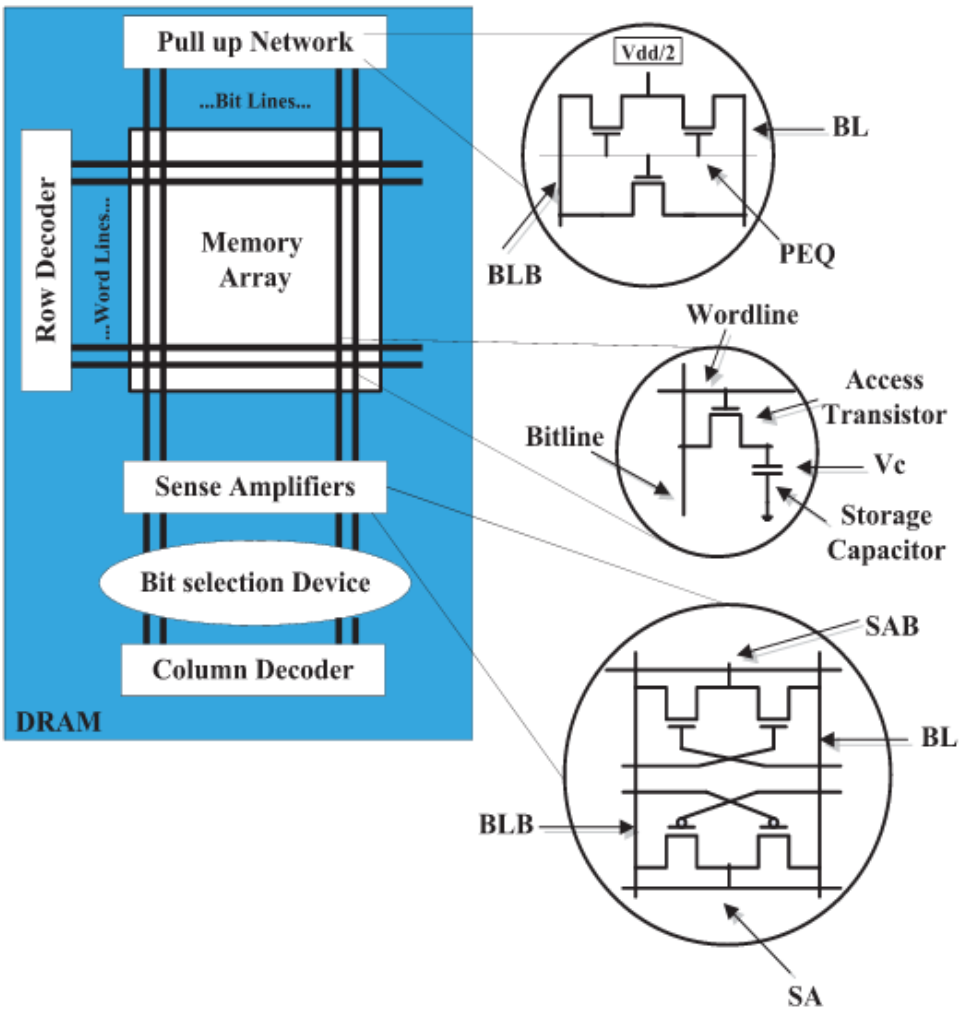

Figura 3 – Arquitetura do DRAM PUF

é possuir um único desafio. Modelos matemáticos podem ser desenvolvidos para predizer as respostas baseado no único desafio do PUF (RÜHRMAIR; SCHLICHTMANN; BURLESON, 2014).

2.4.5.3 *Strong* PUFs

Os *strong* PUFs aproveitam as características de interconexão de portas lógicas para gerar valores únicos e repetíveis, estes tipo de PUFs também podem ser conhecidos como PUFs *delays*. Os valores são obtidos comparando o *atraso* dos caminhos percorridos por um conjunto de *bits* através das portas lógicas. A impedância entre as portas lógicas durante o processo de fabricação determina os caminhos. Ao final de cada caminho, é analisado qual *bit* chega primeiro, determinando assim qual estado (1 ou 0) esse caminho toma. Entre os modelos PUF mais comuns baseados em *delays* estão o Arbiter PUF e o Ring Oscillator (RO) (HERDER et al., 2014; ADAMES; DAS; BHANJA, 2016).

2.4.5.4 Influencias externas nos PUFs

PUFs são componentes eletrônicos que são influenciados por distúrbios no dispositivo. Esses distúrbios são comumente chamados de ruídos, podendo ser internos ou externos. Os principais fatores que afetam PUFs são:

- **Voltagem**: Também conhecido como ruído da fonte de alimentação, cria variações de carga e descarga num componente. As respostas dos PUFs podem ser afetadas com essas variações cada vez que elas são energizadas (WANG; TEHRANIPOOR, 2010). Por exemplo, o comportamento das portas de um chip ou das células de uma SRAM (SONG et al., 2021).
- **Temperatura**: O ambiente onde um dispositivo está localizado pode afetar o seu desempenho. Isso inclui os componentes internos do dispositivo. Variações de temperatura podem contrair ou estender a eletrônica interna de um PUF, afetando também suas respostas (SONG et al., 2021). Como o PUF pode estar em diferentes ambientes, analisar seu comportamento em diferentes temperaturas é útil para aumentar a confiabilidade (LIANG; WEI; LIU, 2020)
- ***Negative Bias Temperature Instability* (NBTI)**: refere-se ao envelhecimento de um componente eletrônico ao longo do tempo (MAES; Van Der Leest, 2014). Quanto mais tempo de uso, é possível encontrar falhas que afetam o desempenho de um dispositivo. Essas falhas afetam a vida útil de uma solução que implementa um PUF (WANG et al., 2020; KROEGER et al., 2020).

2.4.6 Aplicabilidade dos PUFs

Os PUFs podem ser usados em diferentes aplicações, oferecendo diversidades de uso dentro da segurança da informação. A seleção dependerá do tipo de proteção e o custo aceito para aplicar a proteção. Soluções baseadas em memórias tem um custo menor, pelo fato de poder usar os componentes que já existem no dispositivo. Já os PUFs baseados em chip envolvem todo um processo de projeto e produção que pode restringir seu uso a soluções muito específicas.

Anti falsificação: Uma forma comum de inibir a pirataria ou falsificação é colocar marcas em *hardware* ou *software*, por exemplo marcas d'agua (DEY; BHATTACHARYA; CHAKI, 2019; SLEIT; FETAIS, 2018). No entanto, essas marcas podem ser reproduzidas e até removidas (TANHA et al., 2012). A singularidade do PUF permite desenvolver marcas únicas, evitando que esta exista em outro *hardware*. Além disso, a robustez do PUF dificulta deduzir e reproduzir PUFs por meio de um PUF conhecido.

Propriedade Intelectual (PI): Engenharia reversa, espionagem industrial e roubo de projetos de circuitos integrados são alguns dos ataques que existem à PI (VEERANNA; SCHAFER, 2016; XUE et al., 2020). Os ataques podem ser executados pela manipulação de planos de design ou por terceiros por meio das cadeias de desenvolvedores de *software*, componentes e circuitos (HE et al., 2017). Com técnicas baseadas em PUF, é possível dificultar que através das cadeias existam modificações não esperadas no *hardware* ou *software* (SURI; CHAKRABORTY, 2018; LIANG et al., 2016)

Autenticação de dispositivos: Assim como as chaves assimétricas, os PUFs possuem uma parte pública (desafio) e uma parte privada (a resposta), derivada de características físicas obtidas durante o processo de fabricação. Assim, é possível num ambiente controlado criar pares desafio e respostas do PUF, para posteriormente ser armazenadas em um banco de dados (RÜHRMAIR; HOLCOMB, 2014). Já no campo, quando o processo de autenticação é necessário, o dispositivo pode por exemplo, criar uma conexão segura para solicitar um desafio, e usar a mesma conexão para enviar a resposta. O servidor remoto verificará a resposta com os pares armazenados, indicando se a resposta enviada é a esperada.

Armazenamento seguro de chaves: Os métodos criptográficos usam chaves para criptografar e descriptografar conteúdos de forma segura (MUSHTAQ et al., 2017). Manter segura chaves é de vital importância para manter a segurança das informações criptografadas. Comumente, chaves podem ser armazenadas em dispositivos externos como tokens (FOURNARIS; LAMPROPOULOS; KOUFOPAVLOU, 2017), memórias seguras como as *Electrically Erasable Programmable Read-Only Memory* (EEPROM) (MISHRA; BHUNIA; TEHRANIPOOR, 2017), ou usando processadores seguros (SAU et al., 2017). Ataques como análise de energia, falhas de relógio e até mesmo *dump* de memória podem ser realizados como tentativas para obter a chave armazenada (VALEA et al., 2019; COURBON; SKOROBOGATOV; WOODS, 2016). Os PUFs também podem ser entendidos como armazenadores de chaves, com a característica de não haver chave salva, pois a chave é o próprio PUF. Segundo (HERDER et al., 2014), esse conceito é mais seguro, devido que para conhecer a chave, requer manipulações físicas. No entanto, em (ZEITOUNI et al., 2016) é explicado que alguns PUFs podem sofrer alguns ataques não invasivos. Contudo, o PUF mostra-se promissor para armazenar chaves.

3 TRABALHOS RELACIONADOS

Neste capítulo, apresentamos propostas de segurança e métodos de proteção para mitigar vulnerabilidades que afetam a integridade das medições em MI. Os tópicos incluem: proteções assistidas por *hardware*, uso de características físicas do MI, e uso do contexto físico do MI.

3.1 PROTEÇÃO EM MEDIDORES INTELIGENTES ASSISTIDAS POR *HARDWARE* (AHW)

Garantir a integridade e confiabilidade dos dados armazenados e processados por MI é um desafio (HU; WEI; B, 2019). Uma forma de superar o desafio é fornecer ambientes de computação confiáveis onde *software* e *hardware* não sofram alterações por terceiros (ASGHAR et al., 2017). Durante os últimos anos os processadores adotaram recursos de segurança para fornecer ambientes de computação confiáveis, com características como gerenciar chaves criptográficas e fazer distinções de execução entre o sistema operacional, aplicativos, *software*, e *firmware* (SAU et al., 2017; BRASSER et al., 2018). Um dos mais conhecidos é o *Trusted Platform Module* (TPM) (SAU et al., 2017), desenvolvido pela *Trusted Computing Group* (TCG).

Diferentes propostas para MI foram desenvolvidas usando TPM. Em (SIDDIQUI et al., 2018), os autores apresentam um *framework* de segurança que usa TPM para manter a integridade e confiabilidade das informações enviadas por medidores inteligentes. Em (ZHANG; ZHENG; WANG, 2019), os autores, além de estabelecerem comunicações seguras com o TPM, desenvolveram um algoritmo que permite verificar remotamente o estado das chaves criptográficas usando TPM.

Recentemente, Intel e *Advanced RISC Machine* (ARM) deram os primeiros passos para fornecer ambientes seguros em seus CPUs. Intel lançou o Intel *Software Guard eXtensions* (SGX) (COSTAN; LEBEDEV; DEVADAS, 2017; SAU et al., 2017), incorporando em seus processadores um conjunto de extensões responsáveis por construir e gerenciar espaços de memória seguros chamados enclaves. A principal vantagem dos enclaves SGX é a capacidade do processador criptografar a memória, separando de forma segura a execução de partes de código consideradas confidencial (COSTAN; LEBEDEV; DEVADAS, 2017). Similar aos testes caixa preta aplicados no *software*, os enclaves não revelam o seu conteúdo, somente deixam ver as entras e as saídas. Assim, camadas de pilha de *software* como *firmware*, *hypervisor* e o sistema operacional não conhecem

o conteúdo nem tem acesso direto aos enclaves, somente o processador por meio de registradores especiais (BRASSER et al., 2018).

Em (ARAÚJO et al., 2018), os autores apresentam uma proposta para processar dados de MI com segurança, mantendo a privacidade do usuário usando SGX. Cada medidor possui uma chave privada previamente estabelecida que assina os pacotes, permitindo, se necessário, verificar um conjunto de medições provenientes dele. Em (PEREIRA et al., 2018), os autores propõem uma solução para verificar o *software* carregado no MI por meio do Intel SGX. Durante o processo de credenciamento de MI, o código-fonte é submetido e auditado antes de ser carregado. Os autores aproveitam esse processo para adicionar um manifesto assinado pela autoridade metrológica brasileira (Inmetro). Este manifesto é gerenciado por enclaves SGX e apenas pessoas autorizadas podem verificar remotamente o status do software carregado para o MI.

ARM TrustZone é a solução desenvolvida pela ARM para oferecer computação confiável em seus processadores (SAU et al., 2017). O ARM TrustZone cria dois ambientes físicos virtuais independentes denominados "*Safe World*"e "*Normal World*", cada um com seus próprios recursos de memória, controlador, interrupções, entre outros (PINTO et al., 2017). Semelhante ao Intel, o "*Safe world*"atua como um enclave SGX onde fragmentos de código ou aplicativos podem ser colocados lá para serem executados com segurança. A interação entre os dois "*World*"é realizada através de um comunicador de processos denominado *Inter-process Communication* (IPC), encarregado de realizar a operação de acesso aos blocos de memória do "Safe World"e "*Normal World*"(COSTAN; LEBEDEV; DEVADAS, 2017). Em (TENORIO et al., 2019), os autores propõem um medidor inteligente que realiza o processo de leitura de um MI com segurança. O MI implementa um conjunto de funções que captura as informações dos sensores de medição e as criptografa antes de enviá-las a concessionária. Estas ações são realizadas no espaço "*Safe World*" incluído no ARM TrustZone, evitando assim qualquer manipulação das medições no MI.

As soluções acima protegem os dados que são gerados, salvos e processados por um medidor inteligente. No entanto, essas soluções funcionam com base no pressuposto de que os sensores são confiáveis. Também se presume que os canais de comunicação física entre o processador e esses sensores sejam seguros. Em (PALEY; HOQUE; BHUNIA, 2016), os autores demonstraram que no console de jogos Xbox, os barramentos de comunicação são altamente rastreáveis a ataques físicos. Em (LEITÃO; VASCONCELLOS; BRANDÃO, 2014), os autores demonstraram como é possível fazer alterações e manipulação de componentes físicos em bombas de combustível.

3.2 PROTEÇÃO USANDO CARACTERÍSTICAS FÍSICAS (CAF)

Uma característica dos medidores inteligentes (MI) é pertencer a ambientes facilmente acessíveis a terceiros. Essa característica expõe o MI a possíveis violações no *hardware* e *software*. Os PUFs mostram-se como uma forma promissora de utilizar características físicas do *hardware* em métodos de segurança, principalmente em dispositivos com limitações, como MI.

Cargas de *software* não autorizados não são exclusivos em MI; este problema afeta a maioria dos dispositivos inteligentes. Em (LEE; MARKANTONAKIS; AKRAM, 2016), os autores propõem uma técnica para vincular o *software* a um *hardware* por meio de um PUF. Um valor único vindo do PUF é usado pelo processador para criptografar e descriptografar as instruções do *software*. Dessa forma, se um invasor copiar o *software* carregado na memória, apenas o processador com o valor PUF apropriado, executará corretamente as instruções. Os autores em (QIN et al., 2020) propõem uma arquitetura segura para dispositivos inteligentes usando TrustZone e PUF. A arquitetura usa TrustZone para separar com segurança as partes do *software* que precisam maior segurança, como funções de inicialização e verificações de certificado. O PUF é usado para duas finalidades: (i) para gerar uma identidade do *hardware*, usada para assinar o *software*, e (ii) para gerar uma chave que criptografa as instruções de retorno do *software* carregado no "Normal World"pelo TrustZone. A criptografia nas instruções de retorno evita ataques chamados de *Return Oriented Programming* (ROP) (ZHANG; SEKAR, 2015; CLERCQ; VERBAUWHEDE, 2017), caracterizados por desenvolver funções em tempo de execução sem a necessidade de injetar código.

Os medidores de energia inteligentes fazem parte da chamada *Smart Grid*, onde a concessionária e o MI estabelecem comunicações diretas e em tempo real. No entanto, para manter a integridade dos dados enviados por ambas as partes, é necessário estabelecer canais de comunicação seguros. Em (AMAN; CHUA; SIKDAR, 2017), os autores usam os valores de PUFs para implementar um protocolo de autenticação mútua em dispositivos IoT. A autenticação mútua proposta aproveita a função de desafio e resposta do PUF, onde cada medidor é identificado por meio de sua resposta. O protocolo pode ser usado em dois cenários i) quando o dispositivo deseja fazer uma conexão com um servidor. ii) quando dois dispositivos IoT desejam se comunicar. Em (BOYAPALLY et al., 2020), os autores propõem usar PUF para estabelecer uma identidade do MI e usá-la junto com um algoritmo que permite ter um canal seguro entre medidores e concessionaria.

As comunicações internas do medidor inteligente também podem ser protegi-

das através do PUF. Em (STANCIU; MOLDOVEANU; CIRSTEA, 2016), os autores propõem o isolamento baseado em *hardware* entre processadores e periféricos de um *System on Chip* (SoC). SoC é um microchip que integra a maioria dos componentes do computador como: processador, memória e vídeo. O isolamento apresentado pelos autores é baseado em domínios que criptografam a comunicação entre os componentes de um SoC. Dessa forma, os invasores que analisam o canal de comunicação não poderão saber as informações por eles transferidas. O canal seguro é baseado em um algoritmo que cria chaves de criptografia a partir do valor gerado por um PUF. O experimento foi desenvolvido criptografando a comunicação de dois processadores dentro de um SoC. Em (CULTICE; LABRADO; THAPLIYAL, 2020), os autores propõem uma arquitetura para uma troca segura de mensagens entre componentes usando o protocolo para veículos *Controller Area Network* (CAN). Cada componente possui um PUF com o qual gera uma chave privada que é usada como parte do *Elliptic-curve Diffie–Hellman* (ECDH) para gerar chaves assimétricas. A cada nova seção, o PUF é usado para gerar uma nova chave. Uma característica dos PUFs baseados em memória é usar *bits* estáveis para gerar chaves privadas. No entanto, os autores (ECKERT; TEHRANIPOOR; CHANDY, 2017) demonstraram a viabilidade de se obter *True Random Number Generators* (TRNG) por meio de PUFs que podem ser considerados instáveis. Em outras palavras, *bits* que se comportam aleatoriamente. O uso deste tipo de *bits* é útil como sementes para algoritmos criptográficos (GUPTA; GUPTA; SINGH, 2019).

As soluções PUF são usadas para proteger os dados armazenados nas memórias, execuções do processador e criar canais de comunicação seguros. Na comunicação física, eles se mostram promissores, embora ainda estejam nos estágios iniciais. Os trabalhos de (STANCIU; MOLDOVEANU; CIRSTEA, 2016; CULTICE; LABRADO; THAPLIYAL, 2020) mostram a viabilidade de usar PUF para criar canais físicos seguros e os primeiros passos para incluir outros componentes além de memória e processador. Uma das principais limitações para implementar o PUF em componentes internos é a capacidade computacional exigida pelo componente interno. Nosso trabalho apresenta uma forma de estender o uso do PUF para outros componentes internos.

3.3 PROTEÇÃO BASEADAS NO CONTEXTO FÍSICO (CF) DOS COMPONENTES

O contexto físico do dispositivo e as interações com o ambiente ao qual ele pertence é uma alternativa robusta às soluções de identificação e autenticação mais convencionais. Os trabalhos aqui apresentados utilizam o contexto físico como uma forma de aprimorar a segurança dos dispositivos, principalmente inteligentes e IoT.

Exemplos de contextos físicos podem ser luminosidade, temperatura, umidade.

De acordo com (HABIB; LEISTER, 2015), o contexto físico pode ser expandido para características próprias do dispositivo, como rede, uso da memória, inclusive interações com outros dispositivos. O autor também inclui o posicionamento do dispositivo, uso do GPS, informação do *touch screen*, microfones, comportamento do usuário, entre outros. Portanto, pode-se deduzir que o uso de contexto físico em segurança da informação para identificar e autenticar dispositivos inteligentes e IoT é promissor. A seguir, listaremos aqui alguns trabalhos que, por meio do contexto físico, aplicam segurança ao dispositivo e às informações geradas pelo dispositivo.

Em (JIANG et al., 2019), os autores propõem o uso de pulseiras inteligentes com acelerômetros para incluí-los nos processos de pareamento. O movimento das pulseiras durante um aperto de mão é aleatório, sendo robusto o suficiente para ser parte de um algoritmo gerador de chaves únicas. Assim, após movimentar o pulso, cada dispositivo calcula sua chave única com base no contexto físico obtido, compara as chaves e, se forem iguais, pode-se realizar o emparelhamento.

Em (WANG et al., 2018), os autores apresentam a pulseira SecureTag, um dispositivo que aproveita a propagação de ondas de rádio para evitar ataques DoS em dispositivos IoT usados no corpo. Os autores propõem usar o espectro gerado pelas ondas quando atingem o corpo humano, por exemplo, a antena WiFi. O espectro de onda é o contexto físico e é influenciado pela distância entre as antenas da SecureTag e o telefone celular, em relação ao corpo humano. Os autores apresentam que os dispositivos localizados longe do corpo humano criam um contexto físico muito diferente do que os dispositivos que estão próximos. Portanto, dispositivos distantes do corpo humano não são IoTs corporais e possuem alta probabilidade de serem maliciosos.

Em (MELO; MACHADO; CARMO, 2018), eles exploram a autenticação de dispositivos que compartilham o mesmo contexto físico. Eles propõem o uso de propagação de rádio de um sinal para gerar informações de contexto físico, criando assim uma "impressão digital eletromagnética". Entre as contribuições está a proposta de um modelo para obter *bits* únicos com base no contexto físico e o uso do valor único em protocolos de comunicação.

A aplicação do contexto físico interno atualmente não é tão explorada em relação ao externo. Os trabalhos estão focados em analisar componentes *hardware*, coletando dados relacionados ao comportamento, isso pode ser entendido como uma impressão digital interna do dispositivo. Entre os trabalhos encontra-se os que exploram imperfeições no oscilador de cristal, embora possua limitadas imperfeições, é possível realizar identificadores de *software* e *hardware* (SANCHEZ et al., 2021). O uso de *Real*

Time Clock (RTC) e Digital Signal Processor (DSP) são úteis para esta finalidade (SANCHEZ; SANTOS; BALZAROTTI, 2018). O tempo que um dispositivo leva para realizar uma determinada função pode ser usado para modelar o comportamento de um sistema (De Castro et al., 2017). Existem também trabalhos que analisam e monitoram o comportamento de componentes como CPU, memória e comunicação (DONG et al., 2019). Essas propostas limitam-se a dispositivos com capacidade de armazenamento e análise de dados, dificultando seu uso para dispositivos com recursos computacionais limitados. Outro trabalho explora as características internas dos microfones em *smartphones*. Em (BALDINI; AMERINI, 2019) os autores criaram um ruído branco que é captado pelos microfones, onde através de uma Rede Neural Convolucional é possível extrair características únicas de cada microfone.

Os trabalhos anteriores têm em comum a capacidade de capturar um contexto físico, convertê-lo em *bits* e obter valores que podem ser incorporados aos métodos de identificação e autenticação. Eles também têm em comum a capacidade de inferir quando outro dispositivo compartilha o mesmo ambiente. Essa característica é importante ao desenvolver identidades com base em dispositivos que compartilham e observam o mesmo contexto físico.

O contexto físico é uma nova forma de aprimorar os métodos de identificação e autenticação usando as características do ambiente e as características produzidas pelos dispositivos. No entanto, uma parte pouco explorada é a aplicação desses conceitos às interações internas do dispositivo e seus componentes. Nossa pesquisa dá um primeiro passo nesse sentido e mostra como os componentes de um dispositivo têm a capacidade de identificar seu ambiente físico.

3.4 CONCLUSÕES SOBRE OS TRABALHOS RELACIONADOS

Nas subseções anteriores, apresentamos trabalhos que aumentam a segurança em dispositivos com recursos computacionais limitados. A seguir, explicamos as principais diferenças dos trabalhos relacionados e nossas propostas.

Os trabalhos assistidos por *hardware* incluem soluções como TPM, SGX e ARM TrustZone (ver seção 3.1). Embora estas sejam soluções proprietárias, elas são adaptáveis para desenvolver propostas que aumentem a segurança dos dispositivos. Por exemplo, em trabalhos desenvolvidos com TPMs (SIDDIQUI et al., 2018; ZHANG; ZHENG; WANG, 2019), observamos duas principais utilidades: armazenar chaves criptográficas que, por sua vez, possibilitam autenticar o dispositivo; e a geração canais de comunicação seguros. Essas duas utilidades dificultam que terceiros tenham acesso ao

valores de medição, tanto no MI como nas comunicações entre o MI com o servidor. Nos trabalhos que implementam SGX e ARM TrustZone (ARAÚJO et al., 2018; PEREIRA et al., 2018; TENORIO et al., 2019), a tendência está no uso de enclaves do processador (ver seção 3.1) para realizar processos de faturamento e medição, incluindo a verificação da identidade do *software* carregado no MI, inclusive criar uma identidade do processador. Contudo, os trabalhos com TPM, SGX e TrustZone, têm uma característica em comum, não possuem a capacidade de identificar se as medições recebidas e processadas são de uma origem legítima. Em outras palavras, não é possível identificar se os componentes internos do MI, responsáveis pela medição sofreram algum tipo de alteração ou modificação. Isto deve-se ao fato que as proteções estão encarregadas de manter seguro o processador e a memória interna do MI. Em nossa proposta, aproveitamos as características físicas dos componentes que fazem parte do processo de medição e os incluímos como parte da identidade do MI. Se algum componente que faz parte da medição for removido/substituído ou adulterado, a identidade desse MI não será mais a esperada e ações serão tomadas, por exemplo, não aceitar medições desse MI.

No âmbito de soluções que fazem uso de características físicas encontra-se o PUF (ver seção 3.2). Dentro da computação, as soluções PUF são divididas em dois grandes grupos: as baseadas em chips (LEE; MARKANTONAKIS; AKRAM, 2016; STANCIU; MOLDOVEANU; CIRSTEA, 2016; AMAN; CHUA; SIKDAR, 2017; BOYAPALLY et al., 2020), projetadas especificamente para esse fim; e PUFs baseados em memórias voláteis, como SRAM e DRAM (AMAN; CHUA; SIKDAR, 2017; ECKERT; TEHRANIPOOR; CHANDY, 2017; QIN et al., 2020). Em ambos os casos, as soluções tem capacidade de gerar identidades que são contidas por elementos físicos durante o processo de fabricação. Assim como as soluções assistidas por *hardware*, as soluções com PUF projetam a identidades baseadas em um único componente (chip ou memória). Para o caso específico dos MI, existem componentes internos envolvidos no processo de medição, onde através dos PUFs estes podem ser incluídos na identidade. Dessa forma, nossa proposta explora essa capacidade e versatilidade do PUF, incluindo-o em componentes internos para gerar uma identidade. Especificamente, propomos a memória PUF, a escolha é motivada pelo custo econômico envolvido (é possível utilizar memórias existentes no dispositivo), e pela facilidade de incluí-las nas propostas e métodos de segurança.

No caso de soluções que fazem uso do contexto físico (ver seção 3.3), os trabalhos exploram características obtidas do ambiente físico e as incluem nos processos de proteção(JIANG et al., 2019; WANG et al., 2018; MELO; MACHADO; CARMO, 2018). Em outras palavras, as soluções capturam informações do mundo real por meio de sensores incorporados no dispositivo, transformam em digital e as incluem em métodos

de proteção. Por exemplo, autenticar um dispositivo através dos movimentos detectados por giroscópios quando é realizado um *handshake*.

Em contextos físicos internos, é comum encontrar trabalhos que rastreiem referências entre *software* e *hardware* para obter um identificador ou uma identidade. A maioria dos trabalhas chama o resultado de impressão digital ou *fingerprint*. Ciclos de máquina(De Castro et al., 2017; SANCHEZ; SANTOS; BALZAROTTI, 2018), consumo de CPU (DONG et al., 2019), são alguns dos exemplos de contextos físicos internos, com os quais o *software* pode ser identificado com o *hardware*. Há também a possibilidade de representar um dispositivo através das características internas de um componente. Por exemplo, gerar identidades por meio do microfone ou da câmera que inclui um celular (BALDINI; AMERINI, 2019; SANCHEZ et al., 2021). Qualquer componente que possa capturar algo de seu ambiente interno é útil para esses tipos de contextos.

Nossa proposta se baseia nos conceitos do contexto físico, com a diferença que exploramos as características internas do dispositivo. Assim como o mundo externo do dispositivo fornece valores únicos, propomos explorar o ambiente interno do dispositivo para obter valores únicos e aplicá-los em métodos de proteção.

Na Tabela 2 apresentamos de forma resumida as características dos trabalhos relacionados e as diferenças com nossa proposta. Na primeira coluna temos o autor do trabalho, seguidamente os aspectos gerais da proposta, posteriormente desvantagens da proposta, seguido pela informação referente ao experimento, podendo ser teórico e implementado. Teórico refere-se a trabalhos que validam a proposta através de métodos matemáticos, equações e teoremas, entre outros. No caso dos implementados, são realizados experimentos e em alguns casos protótipos. As colunas restantes são os tipos de proteções que as propostas atingem. Na Tabela 5 explicamos o significado de cada sigla das colunas.

Autor	Aspectos gerias	Desvantagens	Tipo Exp	IdAu	CC	Idt	PP	PM	SW	CoI	CFx	CFi	PMd
(SIDDIQUI et al., 2018)	Mantém informação confidencial. Criptografia de ponto a ponto. Chave Criptográfica dentro do TPM. Experimentos com Raspberry PI.	Precisa que todos os nodos incluam um TPM. Não se preocupa pelos componentes internos dos nodos do smart grid.	AHW Implem	x	x								
(ZHANG; ZHENG; WANG, 2019)	Propõem um método para verificar MI. Chave é criada pelo MI e o TTA (trusted test Agent). TPM é usado para verificar a inidentidades de MI. Quando verificada a identidade, é possível criptografar as informações que saem do medidor.	Não existe algum tipo de proteção da memória interna do MI através do TPM. Não faz verificação dos componentes internos do MI.	AHW Teórico	x		x							x
(ARAÚJO et al., 2018)	Estende a segurança do MI até a cobrança. Ajuda manter a privacidade dos dados privados dos usuários. Usa SGX para anonimizar dados do MI.	Assume que componentes internos do MI não são manipulados. Requer de um processador com tecnologia Intel SGX.	AHW Implem				x	x	x				x
(PEREIRA et al., 2018)	Pensado para ambientes hostis. Evita que software adulterado seja carregado no MI. Focado em proteger a informação que fica na memoria. Permite realizar verificação na nuvem do software carregado. Realiza uma comparação do código original com o código carregado no MI.	Requer de um processador que permita ter um SGX. Não verifica a segurança de outros componentes que fazem parte da medição. Focado em proteger a memoria e o processador. Precisa que o MI tenha um SO.	AHW Implem					x	x				

Tabela 1 – Informação dos trabalhos relacionados parte 1.

Autor	Aspectos gerias	Desvantagens	Tipo	Exp	IdAu	CC	Idt	PP	PM	SW	CoI	CFx	CFi	PMd
(TENORIO et al., 2019)	Focado em manter seguro a privacidade dos usuários. Implementaram a solução usando ARM trustZone. Armazena de forma segura as medições do MI. Ele confia nos sensores que adquirem a medição. Ele encripta a informação no secure *world* do ARM TrustZone. Implementado num Raspberry Pi	A sensor que envia os dados pode ser não confiável. Precisa que os MI tenham um tipo especifico de processador ARM para poder aplicar o método proposto. Focado em proteger a memoria e código	AHW	Implem		x								x
(LEE; MARKANTONAKIS; AKRAM, 2016)	Evita que um software seja carregado indiscriminadamente. Baseado em chip. Somente o chip destinado para esse disp. terá a capacidade de entender o código.	Custo para desenvolver ou comprar chip. Limitações nas optimizações aplicadas no *software*.	CaF	Implem				x	x	x				
(QIN et al., 2020)	SRAM PUF e ARM TrustZone. Juntam software e o valor do PUF como método para evitar manipulação de software. Criptografa os endereços de retorno. Focado para IoT. Secure boot	Precisa de um processador com ARM TrustZone para poder aplicar a método proposto. Não esta especificado informação sobre a memória usada.	CaF	Parc Implem	x			x	x	x				
(AMAN; CHUA; SIKDAR, 2017)	Autenticação mutua usando PUF. Método pode ser usado para autenticar IoT e servidores ou entre IoTs.	Não existem experimentos reais, somente propõem o método. Não esta especificado se é baseado em chip ou memória.	CaF	Teórico	x	x	x							

Tabela 2 – Informação dos trabalhos relacionados parte 2

Autor	Aspectos gerias	Desvantagens	Tipo	Exp	IdAu	CC	Idt	PP	PM	SW	CoI	CFx	CFi	PMd
(BOYAPALLY et al., 2020)	Autenticação mutual usando PUF. Implementou o PUF (baseado em chip).	Precisa de um chip para aplicar o método. Não identifica se existe algum componente modificado.	CaF	Implem	x	x	x							
(STANCIU; MOLDOVEANU; CIRSTEA, 2016)	Cifrado para componentes que residem num SoC. Baseado em chip. Permite verificar outros componentes internos do dispositivo.	Aplicável somente em SoC. Tudo foi simulado.	CaF	Simul		x	x				x			
(CULTICE; LABRADO; THAPLIYAL, 2020)	Cifra mensagens entre nodos que usam o protocolo CAN. Criar canais seguros de comunicação. Usa curvas elípticas.	Nao foi implementado. Possível atraso de mensagens não aceitável em mensagens CAN.	CaF	Teórico		x		x						
(ECKERT; TEHRANIPOOR; CHANDY, 2017)	Usam a mesma forma de obter valores dos PUF para identificar memórias instáveis para criar tramas de bits. Os *bits* instáveis são uteis para criar *True Random Number Generator*.	Não é possível saber se é replicável com outras memórias do tipo. Testaram com um memória antiga, q parece não ser muito usada em dispositivos atuais.	CaF	Implem	x	x								
(JIANG et al., 2019)	Cria um método de autenticação baseado no contexto físico gera do pelo *handshake*. Aplicaram o teste com acelerômetros. Eles forjam o seu próprio contexto físico com variáveis difícil de replicar (movimentos físicos de seres humanos).	Não sei que tão aplicável seja em MI.	CF	Implem	x	x	x					x		

Tabela 3 – Informação dos trabalhos relacionados parte 3

Autor	Aspectos gerias	Desvantagens	Tipo	Exp	IdAu	CC	Idt	PP	PM	SW	CoI	CFx	CFi	PMd
(WANG et al., 2018)	Solução aplicável em dispositivos do tipo *wearable*. Exploração interessante de contexto físico pela propagação de ondas no ambiente.	Existe uma terceira parte que encontra-se fisicamente distante do dispositivo principal.	CF	Implem	x	x						x		
(MELO; MACHADO; CARMO, 2018)	Estabelece uma verificação entre dispositivos antes de estabelecer uma comunicação. Informações únicas encontradas no meio de comunicação é usada como contexto físico.		CF	Implem	x	x						x		
Nossa proposta Características Físicas	Estabelece a identidade do MI incluindo componentes internos que fazem parte do processo de medição.		CaF	Implem			x							x
Nossa proposta Contexto Físico	Criamos um CF interno num MI.		CF	Implem	x		x						x	x
(SANCHEZ; SANTOS; BALZAROTTI, 2018)	Implementaram um *fingerprint* para computadores baseado no tempo de execução.	Aplicável somente para computadores	CF	Implem									x	
(De Castro et al., 2017)	Verificar *software* em sistemas inteligentes		CF	Implem									x	
(BALDINI; AMERINI, 2019)	Usa sonidos para gerar identidades obtidas de microfonos.	Precisa ter armazenamento para armazenar os dados do microfone.	CF	Implem									x	
(DONG et al., 2019)	*Fingerprint* do CPU de um computador.	Focado para computadores	CF	Implem				x					x	

Tabela 4 – Informação dos trabalhos relacionados parte 4

Sigla	**Definição**
IdAu	Identificação e Autenticação
CC	Canais de Comunicação
Idt	Identidade
PP	Proteção Processador
PM	Proteção Memória
SW	Software
CoI	Componentes Internos
CFx	Contexto Físico externo
CFi	Contexto Físico interno
PMd	Proteção da Medição

Tabela 5 – Informação de siglas

4 IDENTIDADES FÍSICAS PARA MEDIDORES INTELIGENTES

Neste capítulo, apresentamos um método para gerar identidades de componentes de um Medidor Inteligente (MI). Nós obtemos identificadores de memórias SRAM consideravelmente pequenas usando técnicas PUF. Por meio desses identificadores é possível criar identidades, que por sua vez são úteis para proteger contra ataques à integridade dos componentes dos MI.

4.1 PROPOSTA PARA DESENVOLVER IDENTIDADES DOS COMPONENTES HARDWARE

Dentro da Cadeia Legalmente Relevante (CLR) (ver seção 2.1.4) de um MI, há uma variedade de componentes *hardware* que não podem ser nomeados ou identificados individualmente, o que possibilita ataques de integridade a esses componentes (por exemplo, por substituição (LEITÃO; VASCONCELLOS; BRANDÃO, 2014)). Desenvolver uma identidade que inclua componentes presentes na CLR aumentará a integridade do MI, devido que substituições alteram também a identidade final. Adicionalmente, este tipo de identidade espera-se que aumente significativamente a segurança e a credibilidade dos MI. Nas próximas subseções explicamos nossa proposta para incluir componentes presentes na CLR como parte da identidade de um MI.

4.1.1 Componente de segurança em MI

Os MI, como outros sistemas inteligentes, usam estratégias de proteção para manter seguros componentes específicos, por exemplo, processadores, microcontroladores (MCU) e memórias. Para exemplificar melhor esta estrategia, na Figura 4 apresentamos uma arquitetura geral de um MI de energia. O componente MCU é o responsável pela execução do *software* e também pela comunicação entre os demais componentes internos do MI. Entre os componentes que interagem com o MCU estão:

- Interfaces de comunicação serial para interagir com o mundo externo (por exemplo, USB, RS-422, RS-232);
- Interface *Liquid-crystal Display* (LCD) para exibir informações de medição;
- Memórias EEPROM para armazenar dados e parâmetros de calibração;
- Componentes de segurança para garantir a integridade do aplicativo (por exemplo, *Trusted Platform Modules*);

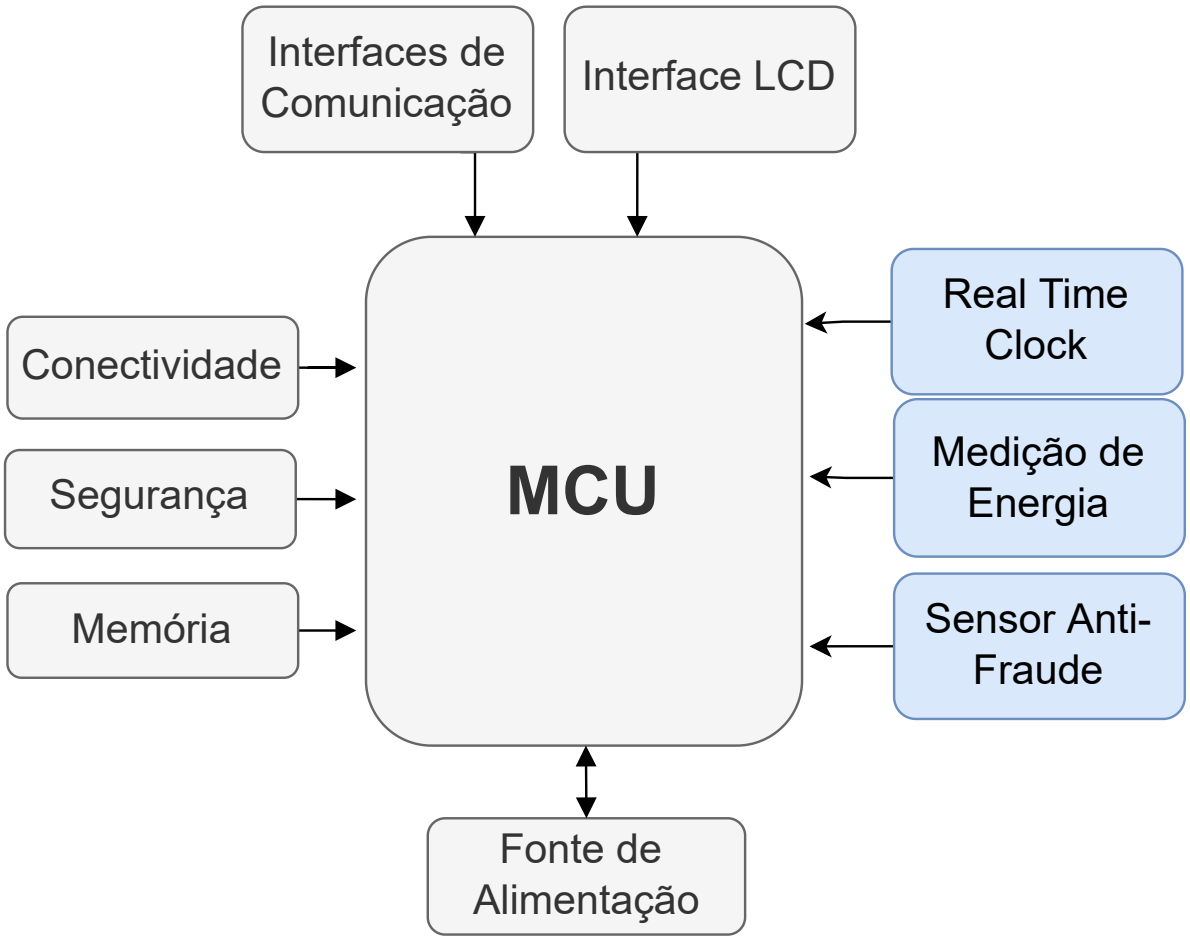

Figura 4 – Modelo de arquitetura baseada no MCU e seus componentes

- O real time clock, para acompanhar a hora atual e trabalhar mesmo que o medidor perca energia;
- O componente de medição de energia que determina as informações de consumo de energia usando quantidades físicas (geralmente tensão e corrente elétrica);
- O componente anti-fraude para saber quando há movimentos bruscos no MI ou aberturas do dispositivo;
- O componente de conectividade para adicionar métodos de comunicação como *Near Field Communication* (NFC), Bluetooth, radiofrequência, entre outros;
- Fonte de alimentação para manter o MI ligado.

Garantir funcionamento correto dos componentes que interagem com o MCU é um desafio. As medições podem ser comprometidas se alguns dos componentes forem adulterados (LEITÃO; VASCONCELLOS; BRANDÃO, 2014). Na Figura 4 definimos componentes em cor azul para representar aqueles que fazem parte da CLR e que podem

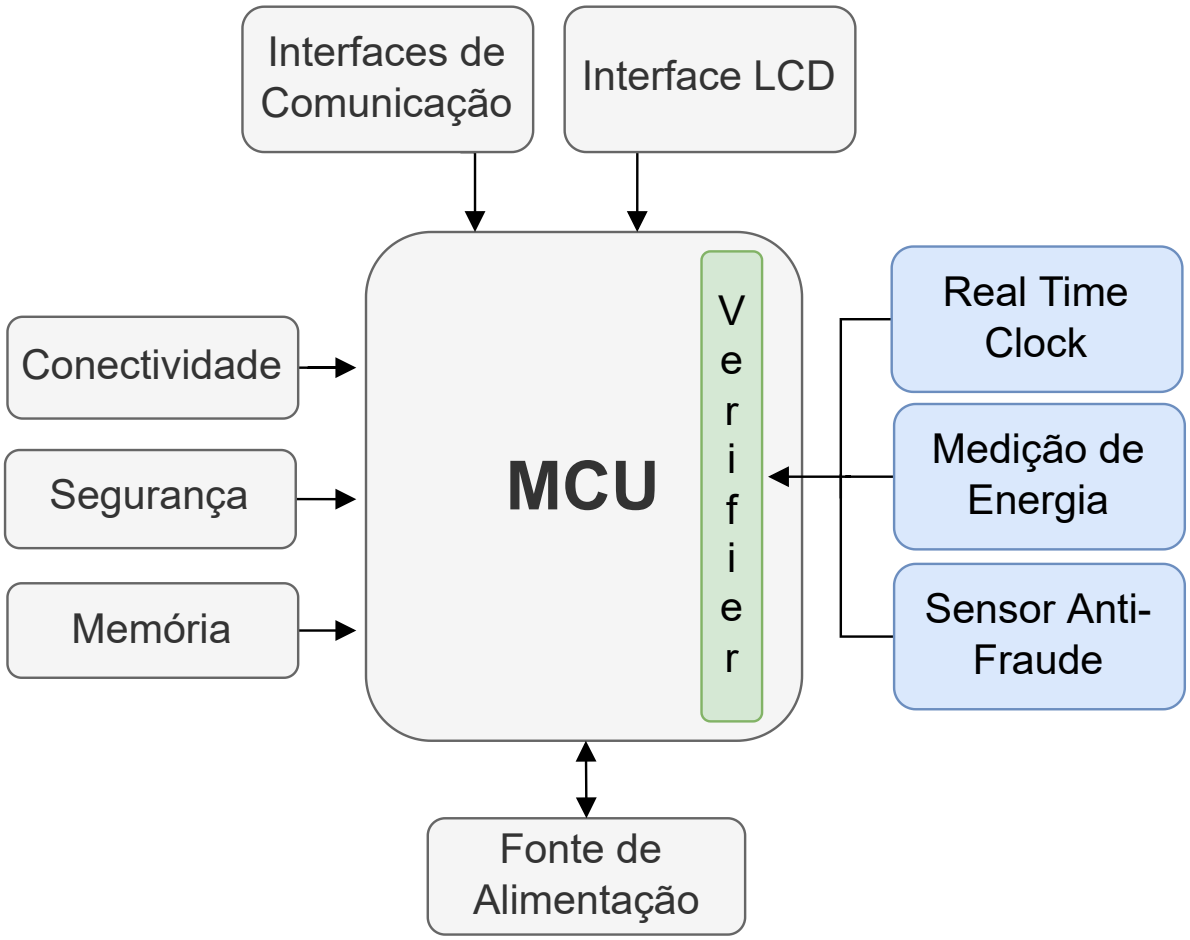

Figura 5 – Modelo de proteção de arquitetura baseada no MCU e seus componentes

ser alvo de manipulações. Observe que o MCU e a memória não foram incluídos na LR, assumimos que esses componentes possuem alguns dos métodos de proteção para MI descritos previamente (seções 3.1, 3.2 e 3.3).

Na Figura 5 apresentamos a arquitetura proposta incluindo o componente chamado *Verifier*, responsável por gerenciar os componentes não incluídos em métodos de proteção existentes para CLR (componentes de cor azul na Figura 5). Através do *Verifier* é possível estender a proteção a componentes que podem comprometer os valores de medição, neste caso, para um MI de energia. A seção 4.1.2 descreve o componente *Verifier* de forma detalhada.

4.1.2 Componente *Verifier* para MI

O mecanismo proposto classifica os componentes em dois grupos: componentes ativos e passivos. Os componentes ativos podem implementar algum tipo de computação (por exemplo, microprocessadores) e, portanto, podemos programá-los para fornecer sua identificação ativamente. Em oposição, um componente passivo não tem tal característica

(por exemplo, memórias), desta forma um componente ativo precisa inspecioná-lo e determinar sua identidade.

Para esta tese propomos o *Verifier* como um componente ativo encarregado de verificar componentes passivos que fazem parte da cadeia LR. Assumimos que o *Verifier* é um componente confiável que não pode ser violado por atacantes. Essa premissa é necessária para confiar nas identidades que ele gera. Deste modo, o *Verifier* usa as informações dos componentes para gerenciar a identidade do MI. As vantagens dessa abordagem são: (i) a capacidade de gerar uma identidade que inclua componentes da CLR dos MI e (ii) detectar e diminuir fraudes baseados na alteração ou manipulação de componentes que fazem parte da LR.

Uma abordagem não coberta por esta tese é ter o *Verifier* fora do medidor inteligente. Neste caso específico, o *Verifier* não precisa ser confiável e pode ser remoto onde, através de um canal seguro, consulta e recria identidades em um servidor externo. No entanto, optamos por ter o *Verifier* dentro do medidor, tornando-o autônomo e não dependente do estabelecimento de canais seguros com dispositivos externos.

Para desenvolver a identidade, propomos também que *Verifier* possa acessar cada componente da CLR do MI para extrair as informações. Como premissa, as informações devem ser únicas e fáceis de recriar quando solicitadas pelo *Verifier*. No entanto, esta informação também deve ser imprevisível (ou seja, difícil de imitar ou falsificar). As características acima são necessários para evitar que um invasor replique ou adivinhe a identidade. Portanto, propomos o uso de técnicas baseadas em *Physical Unclonable Function* (PUF), mais especificamente, SRAM PUF.

Esse tipo de PUF é barato e versátil para obter valores exclusivos e únicos a partir de Características Físicas (CaF). Desse modo, cada componente gera suas informações usando uma pequena SRAM. Este processo é caracterizado por duas fases: extração da identidade e verificação de identidade. Descrevemos os detalhes de ambas as fases nas subseções a seguir.

4.1.3 Processo de extração de identidade através do *Verifier*

A extração de identidade envolve analisar as SRAMs dos componentes e obter informações exclusivas para compor a identidade do MI. Este processo é realizado em tempo de fabricação onde o fabricante inicializa e recupera o conteúdo da memória, através de um *dump* ou mapa de memória. Este processo é realizado várias vezes para posteriormente determinar as posições estáveis da memória. Essas posições na memória são as que mantêm um estado estável (0 ou 1) com alta probabilidade. Selecionamos

essa probabilidade usando um limite de L próximo a 100% para reduzir a incerteza do estado da posição da memória.

Para a identificação de posições da memória estáveis, é necessária uma análise estatística. Assim, propomos a seguinte equação, seja $a = \{m_1(i), m_2(i), m_3(i), ..., m_n(i)\}$ uma matriz de *bits* na mesma posição i em um conjunto n de mapas de memória consecutivos m na mesma posição com *bits* de tamanho k. A decisão se i corresponde a uma posição estável depende das seguintes equações.

$$Q_i = \sum_{i=1}^{n} a(i) \tag{4.1}$$

onde Q_i é a quantidade de 1s na mesma posição i das memórias na matriz a. Q_i é útil para determinar se o *bit* b_i associado à posição i na memória deve ser 0 ou 1:

$$b_i = \begin{cases} 1, & \text{if } Q \geq \frac{n}{2} \\ 0, & \text{caso contrário} \end{cases} \tag{4.2}$$

Finalmente, a seguinte equação representa a probabilidade de estabilidade da posição da memória i:

$$P_i = \frac{Q_i}{n} \tag{4.3}$$

O fabricante executa o processo descrito em todas as posições de memória k. Em outras palavras, para cada local de memória, as equações acima são aplicadas e os valores resultantes armazenados em vetores.

No final, temos dois vetores de tamanho k: o vetor de *bits* esperados $M_b = \{b_1, b_2, ..., b_k\}$ e o vetor de probabilidades de estabilidade $M_P = \{P_1, P_2, ..., P_k\}$. O próximo passo é gerar uma identidade ID_m com base nas duas matrizes. Primeiro, selecionamos todas as posições que atendem ao limite de L em M_P. Essas posições determinam o vetor de *bits* estáveis M_s, de acordo com a seguinte equação:

$$M_s = \{\forall b_i \in M_b | P_i \geq L\} \tag{4.4}$$

Por fim, definimos a identidade ID_m como o conjunto de vetores M_s de cada componente SRAM que integra o processo de identificação. O tamanho de ID_m depende

do número de componentes e de suas identidades. Também podemos representar ID_m como uma sequência única de *bits*, agregando os *bits* estáveis de cada componente SRAM. Fazemos isso usando operações XOR, que são fáceis de implementar no *hardware* e no *software*.

Observe que nada foi mencionado sobre distribuição equitativa de *bits* (0 e 1) da memória. Essa condição é um requisito nas soluções tradicionais SRAM PUF que usam todas as posições de memória. No entanto, nosso objetivo é usar apenas bits estáveis que possam constituir uma identidade, e estes podem estar em qualquer posição da memória. Isso tem um benefício direto no sentido de poder extrair identidades, mesmo em memórias consideradas instáveis.

4.1.4 Processo de verificação de identidade

O *Verifier* executa o processo de verificação de identidade em campo (ou seja, o local de implantação do MI). Na inicialização do sistema, o *bootloader* ou o carregador de inicialização do medidor aciona o componente *Verifier*. O *Verifier* possui uma configuração predefinida que inclui os valores ID_m, assim como a lista de posições estáveis para cada componente. Em seguida, o *Verifier* coleta informações de cada componente que contribui para compor a identidade.

A Figura 6 mostra o processo de solicitação e construção do identificador final. Antes de criar a identidade MI, *Verifier* verifica se as identidades dos componentes individuais estão conforme o esperado. Os valores retornados são armazenados em uma matriz ID_f e comparados com o valor de ID_m armazenado com segurança no MCU.

Se alguma das identidades dos componentes individuais não corresponder ao esperado, o *Verifier* solicitará as identidades novamente. Este processo será repetido 3 vezes no máximo, motivado por dois aspectos principais: 1) os *bits* selecionados para compor a identidade são aqueles com estabilidade próxima a 100%, o que deixa pouca margem de erro e 2) para evitar que o MI entre em um *loop* infinito, o *Verifier* deve definir um conjunto finito de tentativas.

4.1.5 Modelo de ataque

Nos últimos cinco anos, uma investigação intensiva no Brasil expôs uma variedade de ataques contra medidores eletrônicos e também na CLR (LEITÃO; VASCONCELLOS; BRANDÃO, 2014). Esses ataques consideram a manipulação de componentes existentes para alterar o comportamento esperado e afetar o processo de medição.

Assumimos em nosso modelo de ataque que o invasor é uma entidade externa

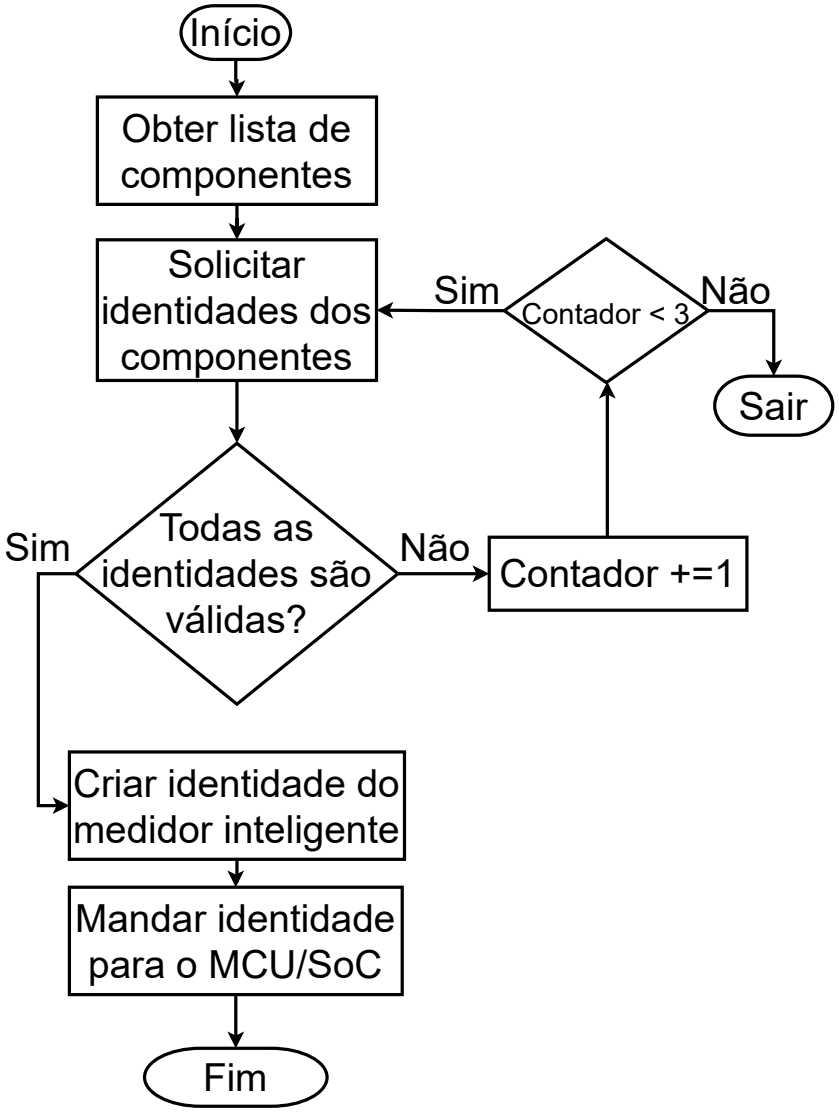

Figura 6 – Processo para verificar identidades

maliciosa (Eve) com acesso ao MI e seus componentes. Portanto, Eve conhece os componentes envolvidos no processo de medição. A continuação, apresentamos os pontos propensos a ataques onde Eve tentará atacar o MI:

4.1.5.1 Componentes

Eve pode alterar um componente que faz parte da CLR do MI. Eve pode executar ataques de *side channel* nos componentes como um método para capturar informações confidenciais. Eve pode adicionar componentes maliciosos para legitimar outros componentes abrangidos. Esse tipo de ataque é mais elaborado e requer amplo conhecimento da arquitetura do MI.

4.1.5.2 Comunicação física

Eve pode analisar o canal de comunicação física (como I2C e SPI) e capturar informações confidenciais para elaborar o ataque. Dessa forma, Eve pode remover um componente legítimo e substituí-lo por um malicioso que realiza o comportamento e as respostas esperadas de um componente legítimo. Portanto, assumimos que o canal de comunicação possui algum método de proteção que impede que esse tipo de ataque ocorra.

4.1.6 Aspectos de segurança

Nesta subseção, discutiremos como nossa arquitetura pode expor os ataques descritos em 4.1.5. Primeiro, assumimos que o MCU e o *Verifier* estão protegidos contra qualquer ataque físico e de *software*. Dessa maneira, o invasor tentará comprometer os componentes que fazem parte da CLR do MI.

Um ataque possível aos componentes da CLR que possuem uma SRAM é o apresentado pelos autores em (ZEITOUNI et al., 2016). Os autores sobre-escrevem partes da memória que é usada pelo PUF, a desenergizam de forma controlada com o intuito de recuperar estados inicias da memória. Desta forma, o atacante identifica uma tendência no estado dos *bits* da memória e através de análises estatísticas, incluindo a voltagem utilizada pela memória, é possível clonar o comportamento e, consequentemente, as respostas do PUF. O ataque aproveita o fato de que a memória SRAM também é usada para armazenar informações. Contudo, esses tipos de ataques não são bem-sucedidos em nossa arquitetura, porque a memória SRAM que faz parte dos componentes não é usada para gravação nem para escrita, sendo esta uma característica necessária para desenvolver o ataque. Em outras palavras, a memória proposta para fazer parte das identidades é usada apenas para ler o estado dos bits quando é energizada. Adicionalmente, o único componente que tem acesso a essa memória é o *Verifier*.

Outro possível ataque é a troca de um ou mais componentes que fazem parte da CLR do MI. Quando isso acontece, o medidor não está mais usando um componente legítimo. Portanto, o *Verifier* identifica que a identidade do MI não coincide com a esperada, devido que, algum dos componentes que faz parte da CLR foi trocado.

Outro ataque consiste na adição elementos maliciosos a componentes legítimos com o intuito de alterar as funções originais do componente. Observe que neste ataque, o componente é legítimo e ainda faz parte da cadeia LR, mas as respostas que ele envia ao MCU ficam comprometidas. Esse tipo de ataque é mais elaborado e requer amplo conhecimento da arquitetura do MI e de seus componentes. Para este caso específico, o

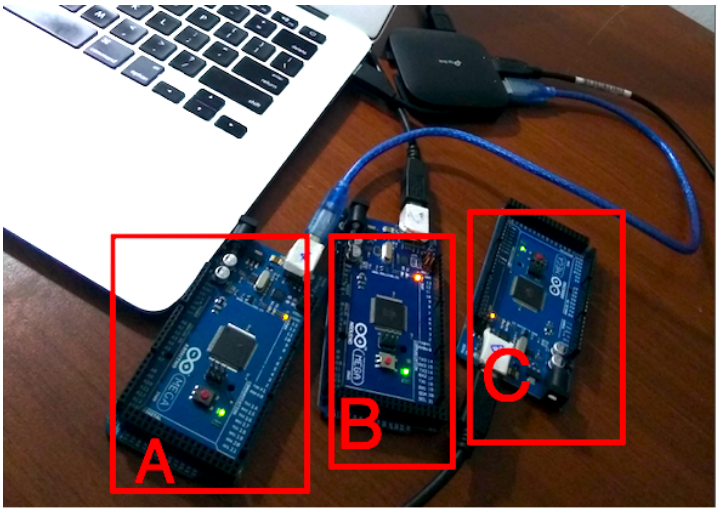

Figura 7 – Esquema de montagem dos Arduino

invasor pode ter sucesso mesmo quando o componente inclui nossa proposta com PUF, devido que, o componente não foi trocado e porque o ataque não necessariamente tenta manipular a SRAM do PUF. Contudo, os elementos adicionados ao componente irão gerar uma impedância que afetará diretamente a inicialização da SRAM e, consequentemente, a estabilidade dos *bits* para geração da identidade. Em outras palavras, se o ataque acontecer, a estabilidade das posições da memória será afetada, gerando valores inesperados que influenciarão a identidade.

4.2 EXPERIMENTOS

Nesta seção, apresentamos os resultados dos experimentos usando memórias SRAM para gerar identidades. Entre os experimentos realizados, verificou-se a viabilidade do uso de SRAM como identidade e também a unicidade entre SRAMs fisicamente idênticas.

4.2.1 Identificação de componentes usando SRAMs de Arduino

O primeiro passo para iniciar aos experimentos é selecionar a memória SRAM. Para os experimentos selecionamos a SRAM equipada no Arduino Mega, com capacidade de 8 KB, obtendo um total de 65.536 bits. Apesar de ser uma memória pequena, sua capacidade é aceitável quando se trata de componentes em MI. Para o experimento utilizamos três dispositivos do mesmo modelo Arduino Mega, assim, cada componente possui SRAM idênticos. Nós os chamamos de Arduinos A, B e C (ver Figura 7)

Implementamos nosso experimento da seguinte forma. Primeiro, ativamos o componente SRAM, de maneira que cada posição na memória assuma um estado

inicial. Em seguida, prosseguimos com a leitura da memória usando um *sketch* [1] que realiza o *dump* de memória e copia seu conteúdo para um computador. Posteriormente, o Arduino é desconectado e a SRAM perde o estado inicial de seus *bits*. Repetimos esse processo 60 vezes para cada dispositivo Arduino. O número de repetições será suficiente para garantir uma distribuição normal do comportamento dos *bits* na memória. No final do processo, temos 180 mapas de memória ou *dumps* (60 mapas para cada dispositivo Arduino). Dividimos esses mapas de memória em dois conjuntos de 30 amostras para cada dispositivo Arduino. Usamos o primeiro conjunto de dados para extrair a identidade SRAM e o segundo para avaliar a eficiência de nosso mecanismo.

4.2.2 Gerando a identidade de fabricação da SRAM

Para o experimento, cada Arduino possui dois conjuntos de mapas da memória (ver seção 4.2.1). Assim, a obtenção da identidade, deve-se primeiro analisar o conjunto de dados das memórias, identificar as posições estáveis e gerar a identidade para os Arduinos A, B e C. Na prática, estamos determinando o valor de ID_m para cada componente (Arduino A, B e C), da mesma forma como um fabricante de medidores realizaria o processo. Em nosso experimento, a análise dos mapas de memória da SRAM foi realizada usando um MacBook Air com processador Intel Core i5 e 4 GB de RAM. Utilizamos os programas Python e Scilab que implementam as estratégias descritas na seção 4.1.3.

O processo de análise de memória começa convertendo o mapa de memória do formato hexadecimal (formato original) para o formato binário. Esse processo facilita a manipulação dos 65.536 *bits* durante a análise. Uma vez que nosso conjunto de dados de análise tenha 30 amostras (isto é, 30 mapas de memória) de cada dispositivo SRAM, podemos estimar a probabilidade de estado de cada posição do bit. Também determinamos o valor esperado mais comum de cada *bit* (ou seja, se a posição avaliada deve ser 0 ou 1).

A Figura 8 exibe um mapa de cores indicando a estabilidade dos *bits* na memória do Arduino A. O mapa de cores representa a probabilidade de mudança das posições dos *bits* em uma região respectiva na memória. Consideramos altamente estáveis os *bits* que mudam de estado abaixo de 10%. Os *bits* que mudam entre 11% e 65% são considerados médios estáveis e o restante dos *bits* são considerados instáveis.

O histograma na Figura 9 mostra a ocupação dos *bits* dentro da memória. Como pode ser observado, os *bits* estáveis correspondem a aproximadamente 70% das posições da memória. Este resultado apoia a ideia de usar o SRAM do Arduino em nosso experimento.

[1] sketch refere-se ao nome do programa usado pelo Arduino.

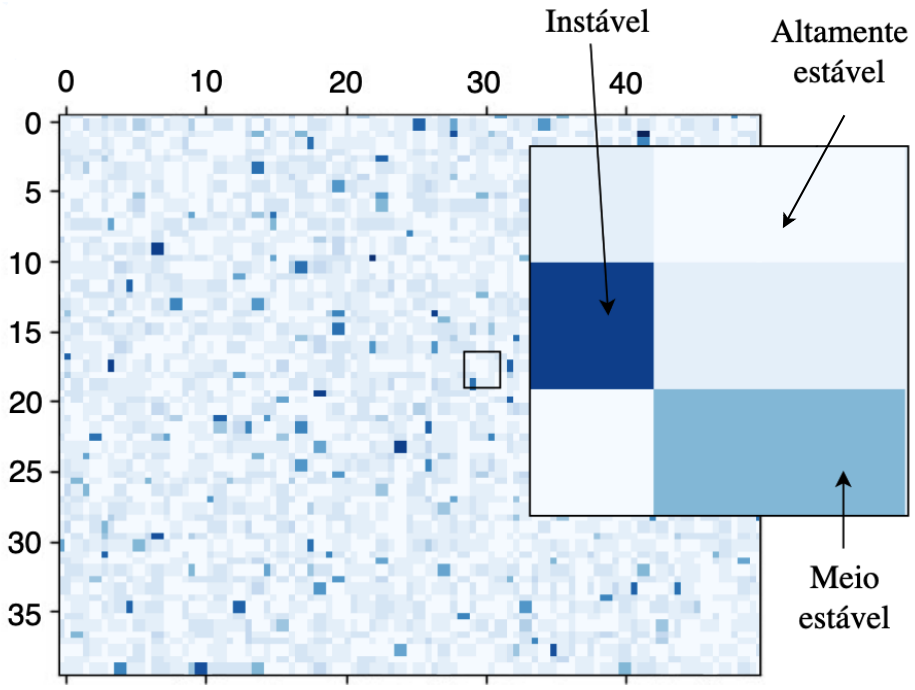

Figura 8 – Exemplo mapa de cores da memoria.

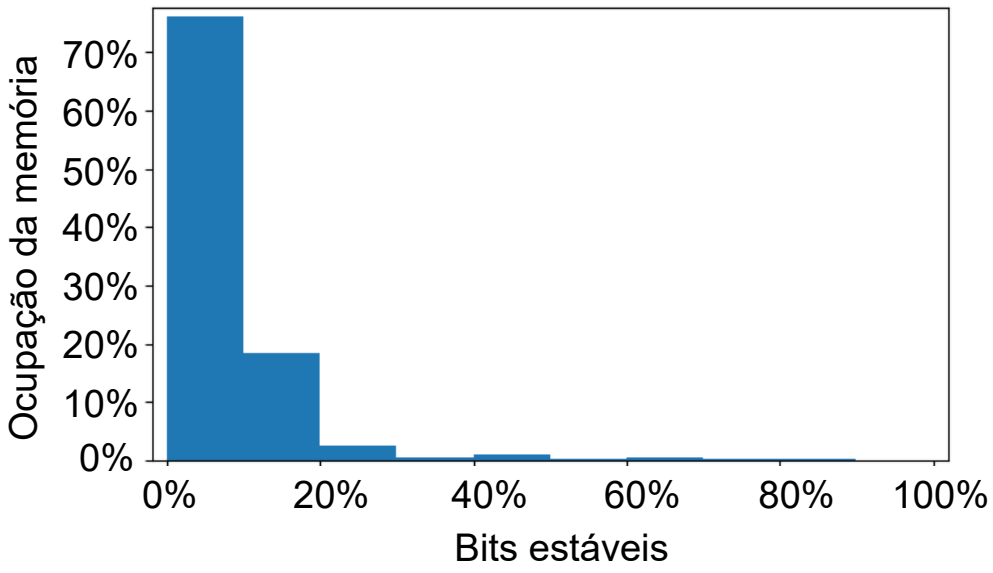

Figura 9 – Histograma ocupação da memória vs *bits* estáveis.

O próximo passo é selecionar os *bits* estáveis e compor a identidade SRAM. Como a porcentagem de *bits* estáveis é alta, podemos usar um limite mais preciso como critério de estabilidade. Nós o definimos como 99%. Quando aplicamos esse limite nas posições de memória, obtemos um total de 58.811 posições estáveis que podemos usar para criar a identidade. A partir desse conjunto de posições estáveis, decidimos usar os primeiros 2.048 *bits* para gerar nossa identidade. Portanto, esse procedimento nos fornece

a identidade ID_m e o vetor de posições estáveis S, para cada um dos três componentes da SRAM do Arduino.

4.2.3 Avaliação de identidade

Nossa proposta depende de dois aspectos importantes: (i) estabilidade da memória e (ii) diversidade entre os estados iniciais dos diferentes componentes da SRAM. Na seção 4.2.2, demonstramos que uma alta porcentagem de posições nas SRAMs dos Arduinos é estável. Usamos esses *bits* para extrair as identidades da SRAM nos Arduinos A, B e C. No entanto, é necessário aprofundar na avaliação para confirmar se a estabilidade é mantida em diferentes leituras da memória. O mesmo processo permite também verificar a diversidade entre as identidades de diferentes componentes. Em outras palavras, quão diferentes são as identidades quando possuem o mesmo tipo de memória SRAM.

Implementamos a avaliação de identidade usando métricas inter HD e intra HD para avaliar PUF (ver seção 2.4.5.1). Atualmente, as implementações com PUF usam esses tipos de métricas para identificar a viabilidade do PUF. O Intra HD permite identificar a possibilidade de obter repetidamente o mesmo valor de *bits* em uma memória. Através desta análise é possível conhecer o comportamento da SRAM quando o *Verifier* solicita a identidade. O Inter HD é útil para saber a diferença entre *bits* de memórias fisicamente iguais usando as mesmas posições da memória. Por fisicamente iguais, nos referimos ao modelo e marca de memória. Assim, com HD é possível determinar:

- Estabilidade da memória para gerar *bits* (Intra HD)
- A exclusividade dos *bits* entre as memórias fisicamente iguais (Inter HD).

Essas duas características são necessárias para obter uma identidade forte e replicável. Na prática, simulamos o componente *Verifier* (ver Seção 4.1.2). Comparamos as identidades ID_m de cada SRAM do Arduino (A, B e C) com diferentes leituras de *bits* estáveis S que obtemos de cada mapa de memória do segundo conjunto de dados. Cada leitura corresponde a uma amostra de uma identidade ID_f que o *Verifier* deve verificar sempre que o MI for iniciado. Esperamos uma alta similaridade na intra HD entre cada SRAM ID_m do Arduino e as amostras de ID_f dos mapas de memória do mesmo componente SRAM. Este resultado deve atestar estabilidade. Complementarmente, esperamos uma baixa similaridade no inter HD entre ID_m e as amostras de ID_f dos mapas de memória de diferentes componentes SRAM. Por sua vez, esse resultado atesta que temos diversidade entre os diferentes componentes da SRAM.

Tabela 6 – Métricas de precisão de identidade usando HD inter e intra.

	Amostras de ID_f		
ID_m	**A**	**B**	**C**
A	0.0005	0.3416	0.3745
B	0.3417	0.0003	0.3822
C	0.3751	0.3826	0.0123

A Tabela 6 resume os resultados da avaliação e confirma nossas expectativas. As células mostram o HD médio entre as identidades ID_m e ID_f. A diagonal principal mostra os valores intra HD, e as células restantes mostram os valores inter HD. Como podemos observar, o inter HD aponta uma alta semelhança entre as identidades comparadas. O Arduino C apresenta o maior valor em termos de distância entre suas identidades. Mesmo assim, esse valor está na ordem de 1%, o que representa alta similaridade. Para reduzir o erro obtido, podem ser aplicados algoritmos de correção de erros, o que permite obter resultados semelhantes aos encontrados nos Arduinos A e B.

Por outro lado, as medidas intra HD indicam uma diferença de mais de 1/3 entre os identificadores dos diferentes componentes SRAM. Esses resultados satisfazem as condições de estabilidade e diversidade. Eles também indicam que podemos estender nossa abordagem em casos práticos envolvendo componentes de medidores inteligentes, como SRAMs e outras memórias. É importante mencionar que quando se fala em PUF, os valores inter HD devem estar próximos de 50% de diferença. No entanto, estamos usando técnicas baseadas em PUF, o que permite alguma flexibilidade na criação das identidades. Uma abordagem para melhorar este aspecto é calcular todas as identidades possíveis de cada componente e selecionar apenas aqueles que atingem cerca de 50% de diferença. Isso é possível porque ainda temos mais de 58.000 *bits* estáveis para formar uma identidade.

4.2.4 Limitações do experimento

Na seção 2.4.5.4 falamos sobre as influências externas que as soluções baseadas em PUF podem ter. A análise dessas influências permite identificar comportamentos esperados pelo *hardware* quando o dispositivo está em uso pelos usuários. A seguir vamos explicar cada uma comparando com nosso experimento:

Temperatura - Para esta tese, os experimentos foram realizados em uma temperatura ambiente entre 24 e 26 graus Celsius. Cobrir maiores faixas de temperatura é o ideal, indo de temperaturas bastante frias como 0 graus Celsius ou menores, a temperaturas maiores que excedem 50 graus Celsius (BOYAPALLY et al., 2020). Para realizar esse tipo de experimento, são necessários laboratórios que possuam *temperature*

chambers, limitando-a a laboratórios especializados para esse fim. Buscas na literatura não revelaram outros experimentos em Arduino que verificassem temperaturas diferentes das que realizamos em nossos experimentos.

Voltagem – Sempre que um dispositivo é ligado, a fonte pode afetar o comportamento dos componentes de um dispositivo (WANG; TEHRANIPOOR, 2010; SONG et al., 2021). Em nossos experimentos usamos uma fonte estabilizada de 5 volts. Embora estabilizado, é normal que as características físicas internas dos componentes afetem seu desempenho. No entanto, essas possíveis alterações tiveram pouco efeito sobre as identidades obtidas da SRAM. Outro experimento encontrado na literatura é aplicar tensões inferiores do que um dispositivo deve operar (ELSHAFIEY; ZARKESH-HA; TRUJILLO, 2017). Isso permite identificar como os métodos propostos são afetados em condições anormais. No entanto, como trabalhamos com medidores de energia, espera-se que essas flutuações sejam mínimas ou no melhor dos casos, nulas, já que os projetistas arquitetam os MI com fontes estabilizadas é incluso baterias para evitar quedas que produzam flutuações na voltagem do medidor.

Negative Bias Temperature Instability (NBTI) - Este tipo de experimento avalia a degradação da vida útil de um dispositivo quando usado de forma intensiva (MAES; Van Der Leest, 2014). Uma vez que os experimentos são projetados para durar dias, semanas, inclusive meses ou anos. Para o caso específico desta tese, seria o uso constante da memória SRAM. Em nossos experimentos não abordamos esse tipo de avaliação, porém, na literatura encontramos os autores (WANG et al., 2020) fizeram um *dump* de uma memória SRAM do Arduino, a fim de conhecer a estabilidade. O experimento durou 2 anos, nos quais foram coletadas 175 milhões de amostras. Os resultados são animadores, durante os experimentos a diferença entre *dumps* da mesma memória foi de 2,49% a 2,97%. Entre memórias os resultados foram ainda melhores, no inicio com 46,79% e no final 46,80%. Embora nesses experimentos foram utilizados Arduino Leonardo (para esta tese o Arduino Mega), este foi o único estudo encontrado na literatura com esse tipo de experimento.

4.3 RESUMO DO CAPITULO 4

Este capítulo apresentou um método para criar identidades com base nos componentes internos de um IM. Problemas de segurança de MI também são abordados, destacando a possibilidade de ataque aos componentes internos, afetando os processos de medição. Adicionalmente, um modelo de ataque foi desenvolvido, explicando como os métodos de segurança existentes não têm a capacidade de detectar alterações de componentes.

Foram discutidos conceitos associados ao uso de CaF e sua influência no desenvolvimento de identidades. Nesse sentido, foi proposto usar PUF como CaF para componentes de um MI e utilizá-lo para gerar identidades. Em especifico, foi selecionado o PUF baseado em memórias SRAM.

Foi proposto um componente de segurança chamado *Verifier*, que verifica as identidades dos componentes pertencentes a CLR de um MI. Para o experimento, foram utilizadas memórias SRAM encontradas no Arduino Mega com 8KB de capacidade. Os experimentos desenvolvidos mostraram que a SRAM do Arduino Mega era robusta o suficiente para incluí-la no processo de identidades proposto.

Por fim, foi apresentado o processo de obtenção e validação de identidades. A avaliação verificou a robustez das identidades, bem como a singularidade entre as memórias SRAM idênticas dos Arduino Mega.

No próximo capítulo será analisado o uso do Contexto Físico como processo para geração de identidades em MI. Especificamente, o ambiente físico compartilhado pelos componentes de um MI.

5 IDENTIDADES BASEADAS NO CONTEXTO FÍSICO

No capítulo anterior, introduzimos o uso de características físicas para gerar uma identidade no MI. Neste capítulo é abordado o uso do Contexto Físico (CF) e sua aplicabilidade na segurança dos MI. A detecção do CF é realizada por meio de dispositivos eletrônicos (como sensores) que capturam informações do mundo físico (analógico) e as convertem em dados (digitais), compreensíveis por sistemas computacionais. Por meio desses dados, é possível implementar estratégias de controle e tomada de decisão.

Deste modo, propomos aumentar a segurança dos MI, identificando componentes *hardware* e suas interações físicas. Mais especificamente, exploramos um aspecto fundamental no *hardware*: o contexto físico dos componentes. Em decorrência, apresentamos um novo mecanismo para desenvolver identidades de componentes em MI. Este mecanismo utiliza o CF para realizar a identificação dinâmica de componentes ativos que compartilham o mesmo ambiente. Entre as contribuições estão:

- Desenvolver um método para obter identidades baseadas no contexto físico dinâmico de dispositivos que compartilham a mesma fonte de alimentação comutada. Como a fonte de alimentação tenta estabilizar continuamente o nível de tensão, as variações de tensão produzem um sinal de variação único e imprevisível. Os componentes que podem descrever este sinal provam que satisfazem as propriedades de colocalização e simultaneidade (ver seção 2.4.3).
- Implementação de uma estratégia de identificação usando um protótipo de *hardware* de Medidor Inteligente (MI), no qual os principais componentes são microprocessadores Arduino e sensores de tensão. Usamos este protótipo para obter resultados práticos que atestam a viabilidade da nossa estratégia de segurança.

5.1 PROPOSTA DE IDENTIDADES BASEADAS EM CONTEXTO FÍSICO

Nesta seção, apresentamos a proposta para proteger os componentes de um MI por meio de identidades baseadas em propriedades físicas. Embora o termo MI possa designar diversos dispositivos que medem diferentes grandezas físicas, nos concentramos em medidores inteligentes de energia.

Para o mecanismo de obtenção de identidades através do contexto físico, utilizamos a definição de tipos e características dos componentes (passivos e ativos) nos

MI propostos na seção 4.1.4. Uma premissa nos componentes ativos que participam da proposta de identificação do CF é compartilhar a mesma fonte comutada. Nossa arquitetura define que esses componentes podem ler o nível de tensão instantâneo no barramento de alimentação de energia. Essa definição é muito plausível em medidores de energia, uma vez que eles precisam medir o nível de tensão para calcular o consumo de energia. Portanto, qualquer medidor de energia possui sensores que podem fornecer essas medições com uma precisão razoável.

Nesse sentido, nosso mecanismo de segurança propõe o uso informações de CF para criar uma identidade de contexto. Normalmente, usa-se identidades baseadas em CF em soluções de autenticação de dois fatores (CONTI; LAL, 2020; MELO; MACHADO; CARMO, 2018). Neste trabalho, também propomos um uso semelhante. Na prática, a verificação de uma identidade baseada em CF no medidor pode expor invasores que tentam adicionar componentes maliciosos ao sistema, ou que tentam personificar ou fingir um componente legítimo usando algum dispositivo externo.

5.1.1 Seleção do Contexto Físico

Em primeiro lugar, é preciso definir um CF, que por sua vez deve estar relacionado a grandezas físicas não previsíveis observadas por todos os componentes do medidor que queremos identificar. Em outras palavras, o CF para esta tese deve estar relacionado a características internas do MI. Isso limita os possíveis contextos físicos aplicáveis. A temperatura pode ser uma candidata, porém, é possível que as variações internas do dispositivo possuam entropia limitada, devido ao ambiente fechado do MI. O mesmo acontece se optarmos pela grandeza física da luminosidade. Sem mencionar que cada componente interno precisará de um sensor para ler a grandeza física selecionada.

Desta forma, a escolha do CF depende do tipo de dispositivo e das respectivas grandezas físicas que mede. Uma vez que este método foca em medidores inteligentes de energia, assumimos a hipótese de poder utilizar o sinal de níveis de tensão do barramento ou comumente chamado de *Voltage Common Collector* (VCC). O uso do VCC torna-se um bom candidato pelas seguintes características:

- Todos os componentes compartilham o barramento e VCC;
- Ocorrências no barramento são observáveis pelos componentes que o compartilham;
- Interações de componentes internos afetam o VCC;
- A fonte que alimenta os componentes influencia o comportamento do barramento;

- Medidores inteligentes de energia já possuem sensores para medir o VCC.

Para usar o VCC como um contexto físico, é necessário que exista algum tipo de unicidade. Esta é uma característica em técnicas que utilizam o contexto físico como parte em métodos de segurança (SANCHEZ et al., 2021). A unicidade permite determinar que o CF seja único em um determinado momento com alta probabilidade.

Desta forma, propõe-se aproveitar as variações que o microcontrolador exerce no VCC ao realizar as operações de medição. As operações estão diretamente relacionadas ao tipo de consumo do usuário, ao *software* e até mesmo aos ciclos da máquina necessários para realizar a medição. Dependendo das operações, será necessária uma maior demanda de energia, que a fonte de alimentação fornecerá de forma não previsível. Como consequência, todos os componentes que fazem parte de um mesmo barramento podem observar as variações exercidas no VCC.

5.1.1.1 Tipo de evento no Contexto Físico

Na seção 2.4.3 apresentamos classificações de contextos físicos de acordo com os eventos que os influenciam. A seguir, apresentamos o tipo de evento associado ao contexto físico proposto. Observe que as flutuações são produzidas no barramento pela fonte de alimentação quando estabiliza o VCC. Assim, consideramos que a causalidade da CF é espontâneo, pois depende diretamente dos comportamentos advindos da fonte de alimentação. Em relação à previsibilidade, podemos considerar que ela é não previsível, pois não podemos determinar o comportamento que a fonte de alimentação terá. Quanto à unicidade, ele é único, pelo fato de não ser possível reproduzir seu comportamento. Por sua vez, a descrição quantitativa, o CF é monovariável, pois é descrito por meio de uma única grandeza física, o VCC.

5.1.2 Mecanismo proposto para obter a identidade

Para esta tese propomos o uso de algoritmos de processamento de sinal. Para processos de identificação com sinais, é comum encontrar Filtros de Médias Móveis (MAF) (ROSTAMI; JUELS; KOUSHANFAR, 2013; MELO; MACHADO; CARMO, 2018), filtros digitais *passa-faixa* (KARAPANOS et al., 2015). O MAF é comumente aplicado a diferentes tipos de sinais, mas é amplamente utilizado em acelerômetros e requer menos processamento para ser aplicado comparado com os filtros de *passa-faixa*. Por sua vez, os processos de verificação são fortemente baseados em medidas de similaridade, como *Função de Coerência* (MAYRHOFER; GELLERSEN, 2007), *correlação de Pearson* (KARAPANOS et al., 2015), *Correlação Cruzada* e *Distância de Hamming* (ROSTAMI; JUELS; KOUSHANFAR, 2013). Assim, optamos pelo uso do MAF para transformar os

sinais obtidos do CF em identidades. Para conhecer a similaridade entre as identidades, optamos pelo uso do coeficiente de *Correlação de Pearson*.

Implementamos nosso mecanismo de segurança adotando a metodologia proposta por (MELO; MACHADO; CARMO, 2018). Consideramos que o *Verifier* é o componente que calcula a referência de identidade baseada em CF, assim a identidade de *Verifier* é definida como ID_V. Por outro lado, um componente ativo *Prover* precisa provar ao *Verifier* que também pode calcular a identidade CF correta representada por ID_P.

Assim, seja $V_t = \{v_{t-n}, ..., v_{t-2}, v_{t-1}, v_t\}$ o vetor que contém as últimas n amostras de tensão do sinal observado no barramento de alimentação de energia. Tanto o *Verifier* quanto um *Prover* legítimo calculam suas identidades de contexto físico no instante t usando a seguinte equação:

$$ID_t = maf(V_t, w) \tag{5.1}$$

onde a função $maf()$ implementa um *Filtro de média móvel* com tamanho de janela w.

Se a identidade do *Prover* (ID_P) for de um componente legítimo, esperamos que $ID_P \approx ID_V$. Portanto, o *Verifier* pode verificar se ID_P satisfaz uma condição de limite, definindo uma função de comparação e verificando a seguinte inequação:

$$1 - abs(correl(ID_P,ID_V)) \leq Th \tag{5.2}$$

onde a função $correl()$ é o *Coeficiente de Correlação de Pearson*, que mede a similaridade entre ID_P e ID_V; a função $abs()$ corresponde ao valor absoluto; e Th é o limite para aceitar esta comparação.

5.2 EXPERIMENTOS

Nesta seção, avaliamos nossa proposta de uso do *Voltage Common Collector* (VCC) como Contexto Físico. Para validar nossa proposta, selecionamos duas plataformas de *hardware*: o Arduino Mega e o Bus Pirate versão 3.6. O Arduino Mega é o mesmo usado no experimento da seção 4.2.1. O BusPirate é um *hardware* de código aberto desenvolvido por Dangerous Prototypes [1] para depurar, analisar canais de comunicação,

[1] http://dangerousprototypes.com

analisar interfaces de comunicação, analisar e programar microcontroladores, entre outros. O BusPirate é baseado no PIC24 com um módulo FT232RL (SSOP) para conectar ao computador via USB. O PIC24 inclui um sensor analógico integrado para ler o VCC. Alguns dos protocolos de comunicação físicos compatíveis com o BusPirate são I2C, SPI e JTAG.

Todas as informações coletadas pelo Bus Pirate e Arduino foram analisadas em um MacBook Air Intel Core i5 com 4 GB de RAM. O *software* usado para analisar todas as informações e dados foi a linguagem de programação Python e o Scilab [2].

5.2.1 Protótipo e configuração

Para este experimento foi realizada uma modificação ao protótipo usado anteriormente na técnica com PUFs no capítulo 4. Para este mecanismo ainda usamos o Arduino Mega, mas também é adicionado o componente BusPirate (BP).

O Arduino Mega funciona de forma semelhante ao primeiro experimento no capítulo 4. Por sua vez, o BP funciona como um sensor e sua função é fornecer informações sobre o CF escolhido. A captura do CF é efetivada através do sensor de VCC incluído no PIC24 do BP. A Figura 10 mostra o esquema do experimento. O Arduino Mega e BP foram definidos para compor componentes ativos capazes de detectar variações de VCC e descrever o CF.

Constituímos dois conjuntos de componentes que representam o *Verifier* e o *Prover*, como proposto na seção 5.1. Uma fonte de alimentação externa estabilizada alimenta todos os componentes. A configuração BP usa os pinos *Analog to Digital Converter* (ADC) [3] e *Ground* (GND) [4] para conectá-los às conexões positivas e negativas da fonte de alimentação usadas pelo Arduino Mega. Na prática, o BP produz medições em tempo real dos níveis de tensão na entrada de energia do Mega. O *Verifier* e o *Prover* realizam leituras do BP a uma frequência de 1 kHz, obtendo aproximadamente 5720 amostras VCC por segundo.

5.2.2 Identidades baseadas em Contexto Físico

A implementação do mecanismo de segurança segue a metodologia descrita na Seção 5.1. Como as identidades baseadas em CF são descritores dinâmicos, as entidades envolvidas devem coletar essas identidades em diferentes momentos. Assim, definimos que cada instância de identidade corresponde a 30 segundos de amostras de

[2] <https://www.scilab.org/>
[3] O termo ADC se refere ao conversor analógico para digital
[4] O termo GND se refere ao aterramento no circuito elétrico

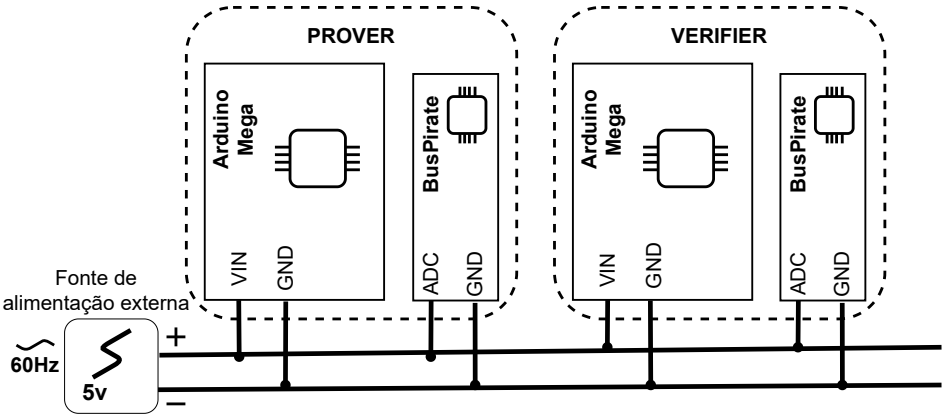

Figura 10 – Mecanismo de segurança #2 esquema com Mega and BP.

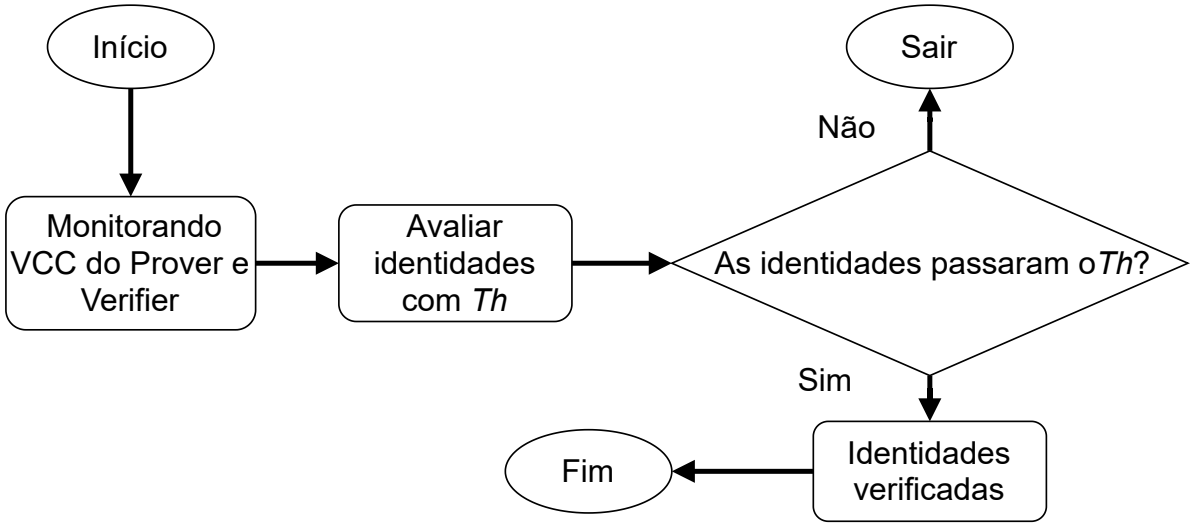

Figura 11 – Processo para verificar identidades usando CF.

VCC, ou seja, as entidades *Verifier* e *Prover* coletam dados brutos de VCC por 30 segundos para calcular cada instância. As entidades usam a equação 5.1 para obter suas identidades, configurando o filtro de média móvel com uma janela $w = 1500$. Esse valor da janela permite ter um alto refinamento do sinal recuperado. Isso é útil ao diferenciar a identidade do ruído que existe no sinal, com janelas menores estaríamos propensos a gerar identidades inválidas.

Verifier e *Prover* foram programados para gerar novas identidades a cada 30 segundos e repetir esse processo até atingir 200 identidades (100 para cada entidade). Dividimos essas amostras em dois conjuntos de dados de 50 identidades consecutivas de cada entidade. Com o conjunto de dados de identidades, podemos verificar a viabilidade de obter identidades baseadas em CF úteis do VCC (ver Figura 11).

5.2.3 Avaliação da influência do microcontrolador sobre o VCC

Nosso primeiro passo é avaliar a correlação entre o VCC e as operações realizadas por um microcontrolador. Por este motivo, um dos Arduinos Mega foi carregado com um *sketch* que faz ao microcontrolador do Mega trabalhar no limite de processamento, chamamos esta ação de *função de estresse*. Basicamente, a função de estresse serve como uma prova de conceito para representar o comportamento de um medidor inteligente de energia.

Este comportamento resulta em maior consumo de energia e, consequentemente, gera flutuações visíveis no sinal VCC. Alternando a execução da função estresse em intervalos de tempo aleatórios, espera-se que o sinal VCC produza entropia, algo necessário para gerar identidades baseadas em CF.

Para validar nossa proposta, é necessário realizar estresse no microcontrolador Arduino. Por estresse, nos referimos a levar o *hardware* a atingir seus limites de desempenho. Um método praticado em computadores para analisar o desempenho são os conjuntos de *software* chamados *benchmarks*, úteis para testar processadores, placas gráficas e memória. No entanto, os *benchmarks* usados em computadores não são compatíveis com a arquitetura encontrada no Arduino. Portanto, foi necessário desenvolver um *sketch* aplicado ao Arduino com base em nossas propostas.

Um aspecto importante do desenvolvimento da função de estresse é conhecer como o microcontrolador do Arduino realiza as operações. Por definição, o Arduino Mega não possui sub-processos ou *threads*. Em outras palavras, o microcontrolador executa uma operação por vez. No entanto, por meio de interrupções, o Arduino tem a capacidade de simular multiprocessamento. Assim, nosso *sketch* deve ser a única operação realizada pelo Arduino, desta forma, forçamos o Arduino a trabalhar em única tarefa ou processo indicado pela função de estresse.

Para desenvolver o *sketch* de estresse, propomos dois estados: estresse e interrupção. O estado de estresse faz com que o Arduino trabalhe em sua capacidade operacional máxima, tanto nas operações do microcontrolador quanto nos acessos à memória. O estado de interrupção, como o nome indica, interrompe qualquer operação realizada pelo Arduino. Para a transição de um estado para outro, foi estabelecido um tempo máximo de 2 segundos e o tempo total do *sketch* é limitado a 30 segundos.

Em outras palavras, os estados se alternam sequencialmente, por exemplo, primeiro estresse, depois interrupção, estresse novamente e assim por diante. A execução de cada estado (estresse e interrupção) deve durar no máximo 2 segundos. Para dar

robustez nos tempos de execução dos estados, foi criado um *array* com valores aleatórios obtidos do site Random.org [5] que utiliza relógios atômicos para gerar os números. A unidade de medida de tempo selecionada é o milissegundo, portanto o *array* contém valores aleatórios entre 0 e 2000.

Definidos os estados do *sketch* e seus tempos de operação, a próxima etapa é selecionar o tipo de operações realizadas pelo *sketch* para estressar o Arduino. Nesse sentido, consideramos dois aspectos: i) a operação deve ser utilizada em conjuntos de *benchmarks* para computadores. Isso garante que seja de fato uma operação aplicada em testes de estresse. ii) estabelecer o ponto de inflexão onde o acesso ao microcontrolador e à memória está em seu limite.

Por conseguinte, optou-se pelo uso de operações de ponto flutuante com números muito grandes. Essa operação faz com que o microcontrolador e a memória do Arduino funcionem no limite. Desse modo, a função de estresse deve gerar picos em relação ao VCC quando for executada.

5.3 CENÁRIOS EXPERIMENTAIS PROPOSTOS

Para realizar os experimentos, propomos dois cenários. O primeiro cenário avalia a hipótese de usar o VCC como um CF. No segundo cenário, analisamos como os componentes de um dispositivo criam identidades a partir do CF proposto. Para obter resultados precisos, várias repetições foram realizadas em cada cenário. Especificamente, 30 repetições para o primeiro cenário e 100 repetições para o segundo cenário.

Em cada cenário, os tempos de repetição foram de 30 segundos. Este tempo de execução é definido para o *sketch* de estresse. Consideramos que o número de repetições é adequado para realizar as análises estatísticas necessárias, uma vez que, a partir de 30 amostras é possível abstrair o conjunto de dados por meio de uma distribuição normal.

5.3.1 VCC como contexto físico

Para validar o VCC como um CF, é preciso verificar se o comportamento do VCC e as execuções do processador Arduino estão correlacionados à execução do *sketch*. Para isso, implementamos o *sketch* proposto na seção 5.2.3.

Durante os experimentos com o *sketch*, a biblioteca padrão do Arduino teve sérios problemas ao trabalhar com pontos flutuantes muito grandes. Consequentemente, o *sketch* proposto não foi capaz de sobrecarregar o microcontrolador o suficiente para ter

[5] <https://www.random.org/>

um efeito no VCC. Para resolver este problema, usamos uma outra biblioteca para realizar operações ponto flutuantes grandes. A próxima etapa foi identificar o limite de operações que o Arduino poderia realizar antes de falhar. Para identificar esse limite, implementamos um *loop* infinito que executa uma operação de ponto flutuante por vez, aumentando seu expoente a cada iteração. Este teste é útil para definir o limite de operações que o Arduino pode realizar antes de parar totalmente. Esse limite é identificado quando o Arduino perde a capacidade de realizar operações de ponto flutuante ou interrompe o microcontrolador.

Para obter medidas mais precisas no experimento, utilizou-se uma conexão estável de energia, evitando assim utilizar a fonte 5V fornecida pela porta USB do computador. Com esta configuração evitamos possíveis oscilações causadas pela porta USB. A nova fonte de alimentação usada é projetada para computadores. Porém, nos experimentos iniciais, não foram registradas correlações entre o VCC e a execução do *sketch* com as operações do ponto flutuante. Análises posteriores estabeleceram que a causa do problema eram: 1) a carga variável da fonte de saída (20 amperes) e 2) o baixo consumo gerado pelo Arduino. Portanto, a carga realizada na fonte pelo Arduino era muito pequena, impedindo gerar picos observáveis pelo sensor. A solução foi usar uma fonte de alimentação para computadores com saída de 5 volts e 2 amperes. Na Figura 12 apresentamos a configuração do esquema com o *Verifier* e *Prover*.

A obtenção das leituras de VCC do Arduino com a fonte de alimentação foi realizada através dos pinos ADC e GNB do BusPirate. O BusPirate foi programado para operar na frequência de 1kHz, sua capacidade máxima, entregando aproximadamente 5720 amostras por segundo. Posteriormente que *Prover* e *Verifier* finalizam suas execuções, eles são conectados à interface USB do computador para baixar e analisar os dados obtidos.

A Figura 13 mostra os resultados deste experimento em duas curvas: uma para o sinal VCC e outra para a presença (ou ausência) da função estresse (*script*). Os resultados demonstram a interdependência entre a oscilação do VCC e o uso da CPU. Deste modo, os resultados também reforçam nossa hipótese sobre o uso dessa grandeza física como um descritor de contexto em MI.

5.3.2 Usando sensores para identificar o contexto físico

A próxima etapa é analisar como os sensores (ou dispositivos) que compartilham o mesmo barramento de alimentação de energia observam o VCC e descrevem o CF (ver Figura 12). Essa análise exige a comparação entre identidades geradas por diferentes dispositivos em um mesmo momento e identidades geradas por um mesmo dispositivo em

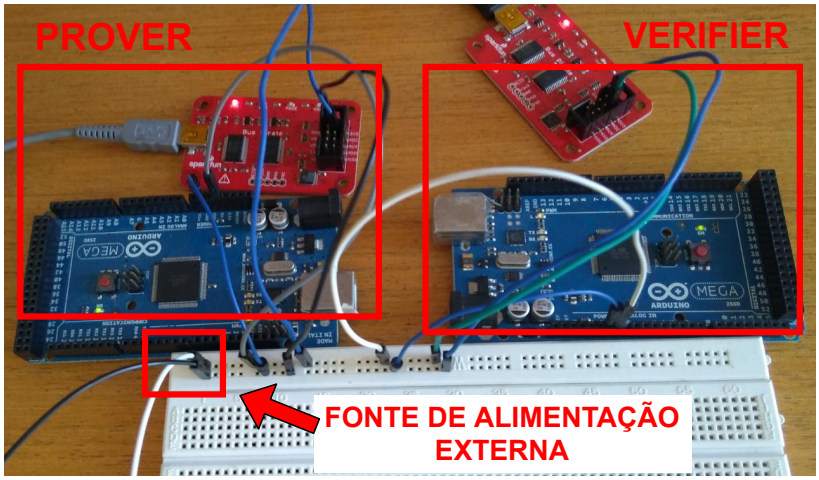

Figura 12 – Configuração do mecanismo #2 proposto

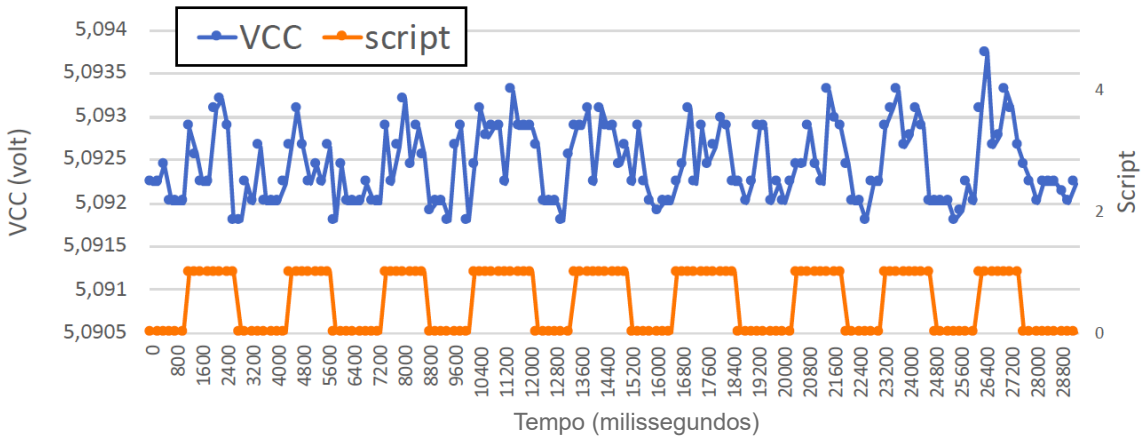

Figura 13 – Comportamento do sinal VCC sob a presença/ausência da função de estresse.

diferentes momentos. Portanto, realizamos a seguinte investigação. Com nosso primeiro conjunto de dados, usamos a Equação 5.2 para comparar cada uma das identidades do *Verifier* com cada uma das identidades do *Prover*. Pode-se esperar que as identidades coletadas simultaneamente apresentem alta correlação, uma vez que descrevem o mesmo CF. Na Figura 14 apresentamos as leituras realizadas pelos pelo *Prover* e *Verifier*. Como esperado, o evento físico capturado pelos dois sensores é o mesmo. Em contrapartida, identidades de diferentes momentos devem apresentar baixa correlação. Na Figura 15 temos o CF observado em momentos diferentes pelo mesmo sensor. Duas curvas VCC são observadas, derivadas da execução da função estresse ao realizar operações e pausas de forma pseudoaleatória. Essas conjecturas são uma premissa básica para validar nossa hipótese inicial sobre o uso do VCC para descrever um CF.

Para avaliar o desempenho das identidades, propomos o uso de métricas

aplicadas em identidades biométricas, em especifico, *False Rejection Rate* (FRR) e *False Acceptance Rate* (FAR). O FRR pode ser entendido como o percentual de casos em que identidades válidas foram rejeitadas incorretamente. FAR é a porcentagem de identidades inválidas que foram aceitas. Basicamente, essas métricas permitem conhecer o nível de aceitação ou negação de uma identidade em um sistema.

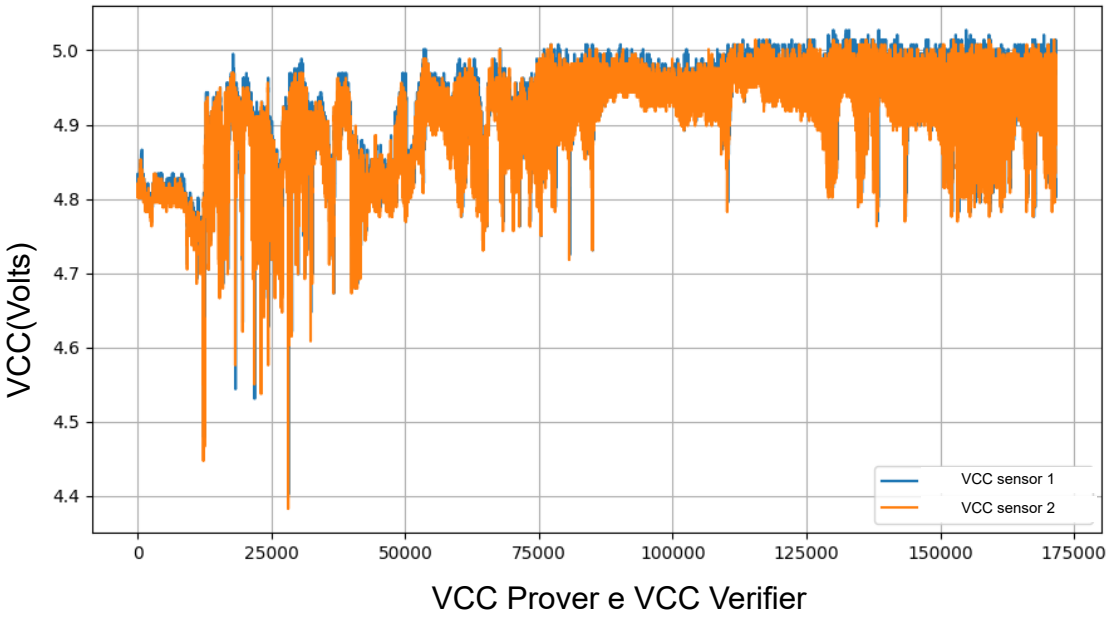

Figura 14 – Representação do mesmo contexto físico observado por dois sensores

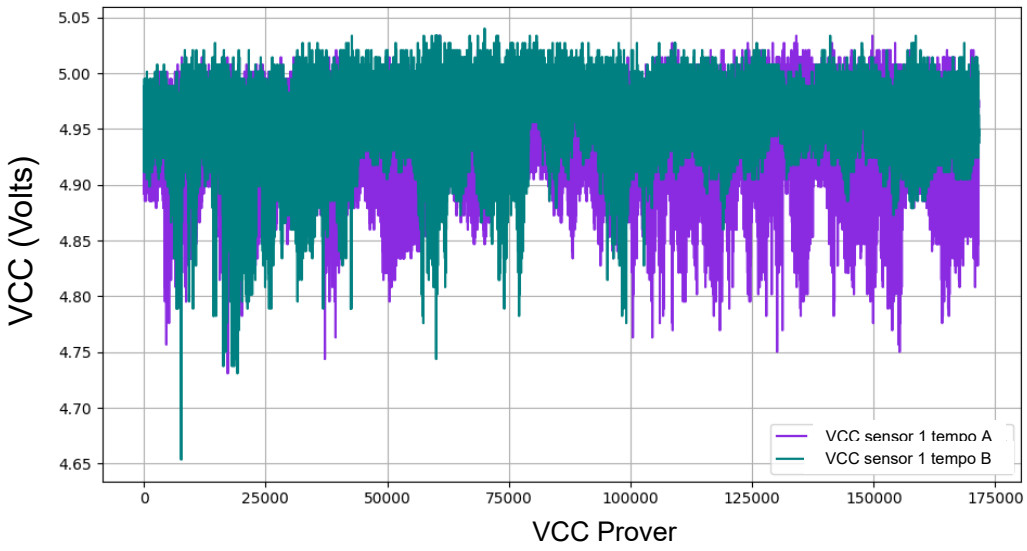

Figura 15 – Contexto físico observado pelo mesmo sensor em tempos diferentes.

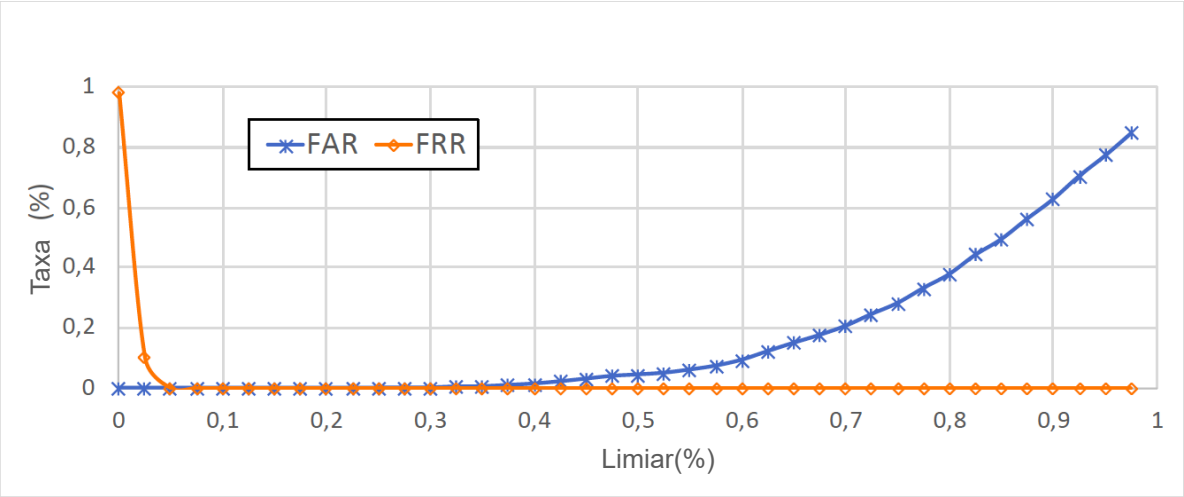

Figura 16 – Como FRR e FAR mudam de acordo ao limiar do valor Th.

A Figura 16 mostra o percentual de FRR e FAR, mesmas métricas usadas por (MELO; MACHADO; CARMO, 2018). O gráfico nessa figura demonstra como FRR e FAR mudam de acordo com diferentes valores-limite de Th. O limite Th determina os critérios para aceitar ou rejeitar uma identidade (ver Seção 5.1).

O objetivo é identificar um valor de Th que minimize tanto FRR quanto FAR. Verificando a Figura 16, pode-se ver que esta condição ocorre quando $0.1 \leq Th \leq 0.3$. Nessas circunstâncias, tanto a FRR quanto a FAR apresentam valores próximos de zero, indicando que podemos comparar diferentes identidades e determinar aquelas que correspondem ao mesmo CF com alto nível de confiança.

5.3.3 Avaliação de identidade

A última etapa é corrigir o valor limite de Th e avaliar a precisão do mecanismo de segurança baseado em CF. Usamos o segundo conjunto de dados para medir nossa precisão na correspondência de identidades. O procedimento é semelhante à análise anterior (ver Seção 5.2.2), mas agora com o valor fixo de $Th = 0.2$. O experimento produz uma matriz de confusão comparando a identidade de cada *Verifier* com a identidade de cada *Prover* no segundo conjunto de dados usando a Equação 5.2. Como o segundo conjunto de dados tem 50 pares de identidades baseadas em CF, a permutação entre as identidades do *Verifier* e as identidades do *Prover* resulta em 1275 comparações, com 50 correspondências corretas e 1225 pares incorretos. A Tabela 7 mostra a matriz de confusão resultante e as métricas de precisão (também conhecida como *valor preditivo positivo*) e acurácia quando $Th = 0{,}2$. O mecanismo de segurança identifica todas as 50 correspondências corretas (sem falsos negativos) e 1223 pares incorretos (com apenas dois falsos positivos). Esses resultados indicam que o mecanismo pode distinguir pares

Tabela 7 – Desempenho do mecanismo de segurança baseado no CF com taxa ($Th = 0.2$).

	OK	NOK
Mesmos IDs	50	0
Diferentes IDs	1223	2

	Metricas
Precisão	0.961
Acurácia	0.998

de identidades baseados em CF legítimos de pares falsos, apresentando alta precisão e acurácia.

5.3.4 Analise de segurança

Nesta subseção discutiremos o mecanismo de segurança proposto. Os ataques são os mesmos considerados no Capítulo 4 (seção 4.1.5). Inicialmente, lembramos que assumimos o MCU e o *Verifier* como componentes seguros. Dessa forma, o invasor tenta comprometer outros componentes que fazem parte do CF.

Nesse sentido, um aspecto de segurança a ser considerado em nossa proposta é a captura do CF por entidades maliciosas. Para capturá-lo, é necessário adicionar um componente malicioso que observe as informações que passam pelo barramento. Um ataque que introduza novos componentes e tente adulterar o medidor, provavelmente precisará fornecer fontes de alimentação individuais (ou baterias). Pode-se esperar essa ação porque os fabricantes projetam a potência original da fonte de alimentação do MI para alimentar apenas componentes legítimos. Adicionalmente, para reduzir a capacidade de sucesso do invasor, o CF é ativado em momentos específicos. Embora esse recurso seja simples, força ao invasor adicionar componentes de armazenamento. Especificamente, precisará colocar um dispositivo ativo dentro do medidor.

5.4 RESUMO DO CAPÍTULO 5

Este capítulo apresentou uma forma de elaborar identidades a partir do CF de um MI. Em primeiro lugar, selecionamos o tipo de CF aplicável ao MI. Como premissa, o CF deve estar relacionado às grandezas físicas não previsíveis do MI e observáveis por todos os componentes identificáveis. Deste modo, foi selecionado a força de tensão existente no barramento do MI.

Para obter as identidades, foi desenvolvido um mecanismo que ao observar o barramento é capaz de criar identidades. Este mecanismo inclui dois componentes, *Verifier* e *Prover*. Ambos componentes monitoram o mesmo contexto físico, mas como o *Prover* não é necessariamente confiável, ele precisa provar que pode descrever o mesmo contexto físico observado pelo *Verifier*. Assim, é de se esperar que os dois possam gerar

identidades bastante semelhantes.

No entanto, para utilizar o barramento como CF, primeiro foi necessário verificar se os componentes *Verifier* e *Prover* observam variações de tensão quando os componentes envolvidos realizam operações que demandam maior ou menor consumo de energia. Assim, nosso primeiro experimento foi criar uma função de estresse com a capacidade de gerar picos de tensão observáveis por *Verifier* e *Prover*. A análise do barramento foi suportada pelo BusPirate, que faz parte dos componentes ativos (*Verifier* e *Prover*) e possui sensor de tensão necessário para descrever o contexto físico.

Posteriormente, foi analisada a correlação do CF observado pelos componentes *Verifier* e *Prover* no mesmo instante de tempo. Essa análise permite determinar quão diferentes as leituras do CF são entre si. Para tal propósito, foi utilizada uma função de correlação. Como esperado, a diferença do CF observado entre os componentes foi mínima. Adicionalmente, avaliou-se a correlação da CF observada pelo mesmo sensor em diferentes momentos. Isso é útil para saber o quão único é o CF proposto. Os resultados foram satisfatórios, a comparação das medições de um mesmo sensor em momentos diferentes, apresentou pouca semelhança entre si.

6 CONCLUSÃO

Após 5 capítulos que inclui a analise das Características Físicas (CaF) e o Contexto Físico (CF) em medidores inteligentes (MI) para desenvolver identidades, este trabalho chega à sua conclusão. De forma geral, pode-se dizer que os objetivos propostos foram alcançados. Porém, é necessário discutir de forma crítica cada um dos objetivos, apontando melhorias e futuras linhas de pesquisa.

6.1 VERIFICAÇÃO DO OBJETIVOS PROPOSTOS

Inicialmente, foram propostos 4 objetivos que delimitaram o desenvolvimento desta tese de doutorado (ver seção 1.2). Esses objetivos são revisados e discutidos nos parágrafos seguintes.

Para atingir o primeiro objetivo, foi apresentada uma base teórica para ajudar a compreender o uso das CaF e do CF na segurança da informação. Foram apresentadas soluções atuais para aumentar a segurança em MI e dispositivos inteligentes. Da mesma forma, foram apresentados os principais ataques em MI, especialmente aqueles baseados em *hardware*.

O objetivo 2 foi alcançado ao apresentar métodos propostos para criar identidades por meio das CaF dos componentes de um MI e do CF observado por esses componentes. Mais especificamente, o Capítulo 3 introduziu o uso de PUF como uma forma de gerar identidades usando CaF. Para o Capítulo 4, foi proposto o uso do VCC como CF, onde os componentes internos de um MI possuem a capacidade de criar identidades. Outra característica dos métodos propostos é a possibilidade de trabalhar em conjunto. CaF gera identidades dos componentes e identidades do CF como um segundo fator de autenticação. Adicionalmente, as duas propostas podem operar em conjunto com técnicas de proteção baseadas em *software* e *hardware*.

Para o objetivo 3, foram realizadas análises para verificar a robustez das identidades obtidas das CaF e CF. Mais especificamente, no Capítulo 3 foram analisadas identidades criadas a partir de uma memória SRAM. Dentre as análises realizadas, verificou-se sua unicidade e independência através da *Hamming Distance*. No caso do CF, no Capítulo 4 foi analisado como os componentes internos de um MI de energia descrevem um CF. A análise incluiu verificar a associação entre o CF proposto e as operações de um microcontrolador. Para auxiliar na verificação, foi desenvolvida uma

função de estresse que adota características de *benchmarks* para computadores. A eficácia da identidade CF proposta foi verificada analisando as porcentagens de *False Rejection Rate* (FRR) e *False Accept Rate* (FAR). A análise também permitiu identificar o valor de limiar ideal para aceitar uma identidade. Os resultados foram promissores e confirmam a viabilidade de usar componentes internos para criar identidades que identificam o CF.

O objetivo 4 foi alcançado através da implementação de protótipos para geração de identidades. No caso da identidade com CaF, foram utilizadas SRAMs incluidas no Arduino para criar PUFs. Para identidades de CF foi usado Arduino e também a placa BusPirate (BP). Os BPs foram usados como sensores capazes de ler o VCC interno do MI. Através da análise das identidades CaF e CF, verificou-se que ambas possuem a capacidade de aumentar a segurança para o MI e seus componentes. Além disso, as identidades propostas têm a capacidade de trabalhar em conjunto com as técnicas de proteção existentes para MI.

A primeira contribuição do nosso trabalho é a apresentação de uma taxonomia para caracterizar os principais ataques a dispositivos inteligentes, abordados sob duas perspectivas: Acesso e ambientes. Por meio dessa taxonomia é possível identificar a complexidade e o conhecimento necessário em *software* e *hardware* para realizar o ataque. No entanto os dois modelos propostos são a principal contribuição desta tese, o que é corroborado pelas publicações obtidas. Em 2020, o método para gerar identidades de componentes por meio de Características Físicas foi apresentado em um congresso internacional. No final de 2020, foi obtida uma publicação em um periódico internacional, onde apresentamos nosso modelo de geração de identidades de Contexto Físico.

Outra contribuição derivada da identidade por meio de características físicas é a capacidade de extrair identidades em memórias consideradas instáveis. Isso é possível porque para gerar a identidade nosso algoritmo não utiliza todo o espaço da memória SRAM, mas realizamos uma análise *bit* a *bit* para determinar a estabilidade.

6.2 SEGURANÇA E LIMITAÇÕES DOS MÉTODOS PROPOSTOS

Nossa proposta tem vantagens para aumentar a segurança dos medidores inteligentes, mas também enfrenta limitações. A seguir, discutimos os principais aspectos para implementar o mecanismo de segurança em medidores inteligentes de energia.

Uma questão crítica é como implementar esses mecanismos sem aumentar o custo dos medidores inteligentes. Dois fatores essenciais devem ser levados em consideração. A primeira é o fato de que ambos os mecanismos exigirão recursos computacionais adicionais. Em muitos medidores inteligentes, os componentes de *hardware* podem ser

superdimensionados. Desta forma, o medidor possui recursos computacionais suficientes para implementar mecanismos de segurança adicionais. Quando este não for o caso, os fabricantes precisarão incluir recursos adicionais que podem aumentar o custo do medidor inteligente.

As decisões de segurança são sempre uma troca entre o preço do produto e os riscos de explorar vulnerabilidades e ameaças. Em outras palavras, onde a fraude de medidores inteligentes é predominante, a adoção de dispositivos mais caros pode ser uma solução razoável.

O segundo fator que deve ser considerado em termos de custo é a adição de componentes relacionados exclusivamente a mecanismos de segurança. Nesse caso, argumentamos que tanto as propostas de Características Físicas (CaF) quanto as propostas de Contexto Físico (FC) não requerem componentes adicionais. O medidor pode extrair identidades baseadas em PUF de suas memórias SRAM internas. Além disso, os medidores inteligentes modernos incluem componentes com memória maior do que os que usamos em nossos experimentos, portanto, é possível ter maior diversidade em termos de identidades.

Por outro lado, as identidades baseadas em CF requerem sensores específicos para capturar informações de CF. Novamente, afirmamos que os projetistas de medidores podem escolher o CF explorando grandezas físicas relacionadas às características do medidor. Nesse sentido, o medidor pode utilizar sensores que já estão em uso para realizar tarefas metrológicas. Este foi o procedimento para selecionar o VCC como CF para o protótipo do medidor inteligente de energia. Se o medidor mede uma grandeza física diferente, por exemplo uma balança, o CF também poderia mudar, explorando as variações na célula de carga da balança.

O tempo de execução de cada mecanismo também é uma questão relevante. O primeiro mecanismo tem duas fases específicas: extração e verificação de identidade (ver seção 4.1.3 e 4.1.2). O processo de extração de identidade não é crítico em termos de execução, pois é executado no momento da fabricação e apenas uma vez. Um longo tempo de execução associado à verificação de identidade seria um problema, mas esse procedimento é executado muito rápido. Podemos justificar essa afirmação com algumas conjecturas sobre nosso experimento. Segundo informações do *datasheet*, o *hardware* Arduino Mega funciona a 8 *bits* e realiza dois ciclos por operação na SRAM. O tamanho de identidade baseado em PUF é de 2048 bits, então o Arduino Mega precisa realizar 256 leituras na SRAM para obter todos os *bits* da identidade. Cada instrução no Arduino Mega requer dois ciclos de máquina, portanto, temos um total de 6 ciclos de máquina por

leitura de SRAM. Assim, em nosso protótipo de *hardware*, o *Verifier* requer uma média de $256 * 6 = 1536$ ciclos de máquina para verificar a identidade de cada componente, o que é pouco tempo mesmo para um microprocessador tão simples quanto o Arduino Mega.

O mecanismo de obtenção da identidade de CF pode exigir mais recursos computacionais para sua execução. Nesse mecanismo, o *Verifier* e o *Prover* (ou *Provers*, já que podemos ter mais de um) precisam fazer as leituras do VCC, filtrar o sinal e convertê-lo em uma identidade. Além disso, o *Verifier* precisa comparar sua identidade baseada em CF com as identidades de cada instância do *Prover* usando uma função de comparação. Essas etapas podem ser restritivas em medidores de energia inteligentes que não possuem recursos. Portanto, consideramos que o mecanismo com CF pode ser limitado a medidores de energia mais complexos, como instrumentos industriais ou sistemas de medição distribuídos.

Em relação à aplicabilidade dos mecanismos de segurança, argumentamos que o mecanismo para identidades CaF pode efetivamente proteger os componentes da cadeia LR em um medidor inteligente. As identidades baseadas em PUF são intrínsecas às propriedades físicas de um componente. Portanto, o *Verifier* deve ser capaz de expor qualquer tentativa de substituir um componente crítico. Eventualmente, um invasor pode tentar ataques mais complexos para contornar esse mecanismo, mas esses ataques também exigirão estratégias mais sofisticadas e, portanto, serão mais caros. Por outro lado, o mecanismo de identidades CF apresenta algumas limitações quanto à sua aplicação. Devemos lembrar que os mecanismos baseados em CF geralmente funcionam como um segundo fator de identificação/autenticação. Pode ser muito eficiente detectar situações em que o invasor tenta substituir outro componente ativo na cadeia LR, e esse componente não conhece o estado atual do CF. No entanto, se o invasor substitui ou adiciona componentes que compartilham a mesma fonte de alimentação sem comprometer as funcionalidades do medidor inteligente, ele também poderá falsificar uma identidade baseada em CF. Assim, embora esse mecanismo possa melhorar a segurança, ele não pode funcionar adequadamente como um mecanismo de identidade primário.

6.3 OPORTUNIDADES DE MELHORIA

Ainda que os objetivos propostos tenham sido atingidos de forma satisfatória, existem aspectos que podem ser aprimorados.

Os experimentos realizados mostram como pequenas memórias SRAM podem ser transformadas em PUFs e gerar identidades através do uso de seus *bits* estáveis. Uma

oportunidade de melhoria é estender a análise dos dados, mais especificamente na unicidade das identidades. Em consequência, novos gráficos podem ser obtidos representando esses novas analises.

Um outro aspecto que pode ser considerado uma melhoria é a exploração dos *bits* usados para gerar identidades. Para contextualizar, a obtenção de identidades é baseada em *bits* considerados estáveis. A porcentagem dos *bits* estáveis influencia diretamente no tamanho e o número de identidades que podem ser obtidas de uma SRAM. Porém, há outro aspecto que pode ser explorado nas SRAMs: a aleatoriedade. A possibilidade de explorar a parte aleatória de uma memória SRAM abre um leque de possibilidades, que *a priori* estariam fora do escopo desta tese. No entanto, consideramos que um passo inicial seria identificar e classificar os *bits* que contêm um nível considerado aceitável de aleatoriedade nas memórias SRAM dos experimentos. Dada a natureza das memórias SRAM utilizadas, que possuem tamanho e capacidade limitadas, é interessante identificar se estas possuem níveis aceitáveis de aleatoriedade, com o intuito de serem aplicadas a algoritmos *True Random Number Generator*. Para identificar a viabilidade de geração de números aleatórios, podem ser aplicadas análises estatísticos (MARTIN et al., 2016) ou mesmo *suites* de estudo de aleatoriedade para esses fins, como a *suite* do NIST (RUKHIN; SOTO; NECHVATAL, 2010).

6.4 TRABALHOS FUTUROS

Nesta subseção vamos apresentar trabalhos futuros que podem ser desenvolvidos através de nossas propostas de identidades com Características Físicas (CaF) e Contexto Físico (CF).

- Desenvolvimento de uma identidade composta por CaF e CF. As identidades propostas nesta tese validaram o potencial de serem aplicadas em dispositivos como medidores inteligentes. No entanto, eles funcionam de forma independente. Um trabalho futuro é desenvolver um método que permita unir as duas identidades em uma. Como resultado, espera-se que a identidade aumente a robustez e a segurança por possuir duas características principais: i) valores intrínsecos dos componentes ii) localização dos componentes com base em um contexto físico. Tal identidade limitaria ainda mais os ataques que buscam manipular componentes internos de um dispositivo.
- Estudo do comportamento da CaF em ambientes controlados. Um aspecto importante de nossa proposta de identidade baseada em CaF é sua aplicabilidade a dispositivos encontrados em vários ambientes naturais. Especificamente, o disposi-

tivo pode ser colocado em ambientes com temperaturas médias, altas e baixas. Essa característica ambiental traz consigo mudanças que podem afetar diretamente a identidade. Portanto, é necessário realizar experimentos em ambientes controlados para memórias do Arduino, analisando como as variações afetam a identidade resultante.

- Estudo comportamental das identidades CaF ao aplicar diferentes nível de tensão de energia. Este experimento permite observar e determinar como as memórias SRAM são afetadas quando essas mudanças existem. Consequentemente, as identidades resultantes também são diretamente afetadas. Esses dados são úteis para determinar variações da identidade e aplicar filtros para corrigir erros se for necessário.

- Aumentar a diferença percentual entre as identidades é outro trabalho futuro possível. Nos experimentos foi constatado que o valor inter-HD está entre 33% e 38%, embora esses valores sejam suficientes para gerar as identidades, seria interessante torná-las mais robustas. Portanto, uma análise das identidades resultantes deve ser realizada e a seleção de *bits* refinada. Consequentemente, espera-se que o valor inter-HD aumente.

- Criação de um método para armazenamento seguro. Existe a possibilidade de usar as identidades CaF e CF como chaves criptográficas para dispositivos com memórias consideradas não confiáveis. Manter os dados seguros armazenados na memória é um requisito de segurança para qualquer dispositivo. O uso de chave pública é a maneira eficiente de contornar esse requisito. Basicamente, a chave pública é usada para criptografar os dados da memória e somente quem possui a chave privada pode descriptografar os dados. No entanto, o uso de chave pública pode ser restrito para dispositivos com recursos computacionais limitados. Assim, é normal encontrar dispositivos que utilizem somente chaves privadas. O desafio no uso de chaves privadas está em como e onde armazená-las sem usar métodos *hardcode*. Organizações como o *Open Web Application Security Project* (OWASP) recomendam o uso de duas chaves privadas. A chave *Data Encryption Key* (DEK) para criptografar os dados e *Key Encryption Key* (KEK) para criptografar a chave que criptografa os dados. Para que este modelo de armazenamento de chaves seja seguro, é necessário que: i) a KEK não seja armazenada junto com a chave DEK, ii) KEK não seja facilmente acessada por um atacante. Deste modo, seria interessante pesquisar o uso dos valores obtidos das CaF ou do CF para criar uma chave do tipo KEK, devido que, a chave não estaria armazenada em lugar nenhum, a chave seria os componentes físicos do dispositivo.

- Investigação para obter identidades CaF em *System on Chip* (SoC). Os SoCs são chips integrados que incluem elementos necessários para compor um sistema

computacional, como processador, memória, vídeo, placas de rede, entre outros. Portanto, seria interessante analisar a SRAM incluída em um SoC e verificar seu potencial como parte da criação de identidades. Da mesma forma, seria interessante avaliar as características aleatórias que a SRAM de um SoC possui. Dependendo do nível de aleatoriedade e do número de bits, eles podem ser úteis como parte de algoritmos criptográficos e sementes para criar números aleatórios.

- No caso do contexto físico, seria interessante estudar outros algoritmos e técnicas de filtragem de sinais. Na tese usamos filtro de média móvel (MAF), porém é possível usar outros filtros para extrair a identidade, por exemplo técnicas com convolução. Assim será possível extrair informações e características de um sinal, inclusive, seria possível extrair distúrbios que ocorrem no contexto físico. O distúrbio do sinal poderia ser também estudado, por exemplo, para identificar alguma manipulação ao componente.

7 PUBLICAÇÕES DO AUTOR

Controlling Smart Meters Integrity via Identity Management of its Components

2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)

DOI: 10.1109/I2MTC43012.2020.9129574

Abstract: This paper presents a security strategy to protect smart meters against attacks that compromise critical software and hardware components. Many of these attacks consist of replacing or tampering memory components that store measuring procedures and sensitive configuration parameters. Our strategy makes use of physical properties from these components to create a secure identity for the meter. Firstly we explore concepts related to Physical Unclonable Functions to extract unique identifiers from SRAM components. Then we combine these identifiers to create a strong identity. Since this identity depends on intrinsic physical features from the meter components, we can implement a mechanism to detect the attacks described before. We also validate our approach in an experiment using a meter prototype built in an Arduino device and connected to SRAM memories. The results show that our idea is suitable for implementation in smart meters as so as in similar embedded devices.

Securing Smart Meters Through Physical Properties of Their Components

IEEE Transactions on Instrumentation and Measurement (Volume: 70)

DOI: 10.1109/TIM.2020.3041098

Abstract: This article presents a security strategy to protect smart meters against attacks that compromise critical electronic components. Many of these attacks consist of stealing energy by manipulating smart meters' measurements. For example, a malicious entity can replace or tamper with components that execute measuring procedures and store sensitive parameters. This type of attack is common on devices, such as smart meters, that are part of environments considered hostile, where attackers have easy access to the device. Therefore, it is necessary to develop techniques in which physical manipulations on smart meters by attackers are unsuccessful. Our strategy

uses physical properties from these components to create secure identities for the meter. We present two main contributions. The first one is inspired by physical unclonable functions and extracts unique identifiers from SRAM components. Then, we combine these identifiers to create a strong identity. The other contribution uses physical context information from the voltage levels in the smart meter's power supply to yield dynamic context identities. We also validate our proposals in experiments using a hardware prototype that embeds Arduino microprocessors, SRAM memories, and voltage sensors. The results show that our idea is suitable for implementation in real smart meters and can help protect these devices from attacks against their components.

REFERÊNCIAS

ADAMES, I. A. B.; DAS, J.; BHANJA, S. Survey of emerging technology based physical unclonable funtions. In: IEEE. **2016 International Great Lakes Symposium on VLSI (GLSVLSI)**. [S.l.], 2016. p. 317–322.

AHMADVAND, M.; PRETSCHNER, A.; KELBERT, F. A taxonomy of software integrity protection techniques. In: **Advances in Computers**. [S.l.]: Elsevier, 2019. v. 112, p. 413–486.

AHMED, C. M.; MATHUR, A. P.; OCHOA, M. Noisense print: Detecting data integrity attacks on sensor measurements using hardware-based fingerprints. **ACM Transactions on Privacy and Security (TOPS)**, ACM New York, NY, USA, v. 24, n. 1, p. 1–35, 2020.

AKDUR, D.; GAROUSI, V.; DEMIRÖRS, O. A survey on modeling and model-driven engineering practices in the embedded software industry. **Journal of Systems Architecture**, Elsevier B.V., v. 91, n. June, p. 62–82, 2018. ISSN 13837621.

ALLADI, T. et al. Consumer IoT: Security Vulnerability Case Studies and Solutions. **IEEE Consumer Electronics Magazine**, v. 9, n. 2, p. 17–25, 2020. ISSN 21622256.

AMAN, M. N.; CHUA, K. C.; SIKDAR, B. Mutual Authentication in IoT Systems using Physical Unclonable Functions. **IEEE INTERNET OF THINGS JOURNAL**, v. 4, p. Tehranipoor, Mark M, 'Mohammad Tehranipoor Mohamma, 2017.

ARAÚJO, M. V. M. et al. Secure Cloud Processing for Smart Meters Using Intel SGX. **SBSeg**, v. 18, p. 89–96, 2018. Disponível em: <https://sol.sbc.org.br/index.php/sbseg/article/view/4274/4205>.

ASGHAR, M. R. et al. Smart meter data privacy: A survey. **IEEE Communications Surveys and Tutorials**, v. 19, n. 4, p. 2820–2835, 2017. ISSN 1553877X.

BALDINI, G.; AMERINI, I. Smartphones identification through the built-in microphones with convolutional neural network. **IEEE Access**, IEEE, v. 7, p. 158685–158696, 2019. ISSN 21693536.

BARKER, E.; ROGINSKY, A. **Transitioning the Use of Cryptographic Algorithms and Key Lengths**. Gaithersburg, MD, 2019. 17–18 p. Disponível em: <https://doi.org/10.6028/NIST.SP.800-131Ar2>.

BATINA, L. et al. In hardware we trust: Gains and pains of hardware-assisted security. In: IEEE. **2019 56th ACM/IEEE Design Automation Conference (DAC)**. [S.l.], 2019. p. 1–4.

BERTINO, E. et al. **Internet of Things (IoT) Smart and Secure Service Delivery**. [S.l.]: ACM New York, NY, USA, 2016.

BETTAYEB, M.; NASIR, Q.; TALIB, M. A. Firmware update attacks and security for IoT devices survey. **ACM International Conference Proceeding Series**, 2019.

BOCCARDO, D. R. et al. Software evaluation of smart meters within a legal metrology perspective: A Brazilian case. **IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT Europe**, IEEE, p. 1–7, 2010.

BOCCARDO, D. R. et al. Software validation of medical instruments. **2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA)**, n. October 2015, p. 1–4, 2014. Disponível em: <http://ieeexplore.ieee.org/document/6860090/>.

BORRA, S. R.; REDDY, G. J.; REDDY, E. S. A broad survey on fingerprint recognition systems. In: IEEE. **2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)**. [S.l.], 2016. p. 1428–1434.

BOYAPALLY, H. et al. Safe is the new Smart: PUF-based Authentication for Load Modification-Resistant Smart Meters. **IEEE Transactions on Dependable and Secure Computing**, v. 5971, n. c, p. 1–18, 2020. ISSN 19410018.

BRASSER, F. et al. Special Session: Advances and Throwbacks in Hardware-Assisted Security. **2018 International Conference on Compilers, Architecture and Synthesis for Embedded Systems, CASES 2018**, IEEE, p. 1–10, 2018.

CAI, L. Z.; ZUHAIRI, M. F. Security challenges for open embedded systems. **2017 International Conference on Engineering Technology and Technopreneurship, ICE2T 2017**, v. 2017-January, p. 1–6, 2017.

CAMARINHA-MATOS, L. M. et al. Technological innovation for cloud-based engineering systems: 6th IFIPWG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2015 Costa de Caparica, Portugal, April 13-15, 2015 Proceedings. **IFIP Advances in Information and Communication Technology**, v. 450, p. I–II, 2015. ISSN 18684238.

CHEN, K. et al. Internet-of-things security and vulnerabilities: Taxonomy, challenges, and practice. **Journal of Hardware and Systems Security**, Springer, v. 2, n. 2, p. 97–110, 2018.

CHEN, K. et al. Internet-of-Things Security and Vulnerabilities: Taxonomy, Challenges, and Practice. **Journal of Hardware and Systems Security**, Journal of Hardware and Systems Security, v. 2, n. 2, p. 97–110, 2018. ISSN 2509-3428.

CHHETRI, S. R. et al. Manufacturing Supply Chain and Product Lifecycle Security in the Era of Industry 4.0. **Journal of Hardware and Systems Security**, v. 2, n. 1, p. 51–68, 2018. ISSN 2509-3428.

CHHETRI, S. R. et al. Security trends and advances in manufacturing systems in the era of industry 4.0. **IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD**, IEEE, v. 2017-Novem, p. 1039–1046, 2017. ISSN 10923152.

CLERCQ, R. de; VERBAUWHEDE, I. A survey of Hardware-based Control Flow Integrity (CFI). v. 1, p. 1–27, 2017. Disponível em: <http://arxiv.org/abs/1706.07257>.

CONTI, M.; LAL, C. Context-based Co-presence detection techniques: A survey. **Computers and Security**, Elsevier Ltd, v. 88, p. 101652, 2020. ISSN 01674048. Disponível em: <https://doi.org/10.1016/j.cose.2019.101652>.

COSTAN, V.; LEBEDEV, I.; DEVADAS, S. Secure processors part I: Background, taxonomy for secure enclaves and intel SGX Architecture. **Foundations and Trends in Electronic Design Automation**, v. 11, n. 1-2, p. 1–248, 2017. ISSN 15513947.

COURBON, F.; SKOROBOGATOV, S.; WOODS, C. Reverse engineering flash eeprom memories using scanning electron microscopy. In: SPRINGER. **International Conference on Smart Card Research and Advanced Applications**. [S.l.], 2016. p. 57–72.

CULTICE, T.; LABRADO, C.; THAPLIYAL, H. A PUF based CAN security framework. **Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI**, v. 2020-July, p. 602–603, 2020. ISSN 21593477.

DABBAGH, M.; RAYES, A. Internet of things security and privacy. In: _____. **Internet of Things From Hype to Reality: The Road to Digitization**. Cham: Springer International Publishing, 2019. p. 211–238. ISBN 978-3-319-99516-8. Disponível em: <https://doi.org/10.1007/978-3-319-99516-8_8>.

DAVIDSON, J. W. et al. A System For The Security Protection Of Embedded Binary Programs. 2016.

De Castro, C. G. et al. EVINCED: Integrity verification scheme for embedded systems based on time and clock cycles. In: **15th IEEE International Conference on Pervasive Intelligence and Computing**. [S.l.: s.n.], 2017. p. 788–795. ISBN 9781538619551.

DEMIGHA, O.; LARGUET, R. Hardware-based solutions for trusted cloud computing. **Computers and Security**, Elsevier Ltd, v. 103, p. 102117, 2021. ISSN 01674048. Disponível em: <https://doi.org/10.1016/j.cose.2020.102117>.

DEY, A.; BHATTACHARYA, S.; CHAKI, N. Software watermarking: Progress and challenges. **INAE Letters**, Springer, v. 4, n. 1, p. 65–75, 2019.

Dhanesh Menon, V. et al. Cyber Security for Smart Meters. **IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing, INCOS 2019**, IEEE, p. 1–5, 2019.

DONG, S. et al. CPG-FS: A cpu performance graph based device fingerprint scheme for devices identification and authentication. **Proceedings - IEEE 17th International Conference on Dependable, Autonomic and Secure Computing, IEEE 17th International Conference on Pervasive Intelligence and Computing, IEEE 5th International Conference on Cloud and Big Data Computing, 4th Cyber Science and Technology Congress, DASC-PiCom-CBDCom-CyberSciTech 2019**, IEEE, p. 266–270, 2019.

EAGLE, C. **The IDA pro book**. [S.l.]: no starch press, 2011.

EBERT, C.; JONES, C. Embedded software: Facts, figures, and future. **Computer**, v. 42, n. 4, p. 42–52, apr 2009. ISSN 00189162. Disponível em: <http://ieeexplore.ieee.org/document/5054871/>.

ECKERT, C.; TEHRANIPOOR, F.; CHANDY, J. A. DRNG: DRAM-based random number generation using its startup value behavior. **Midwest Symposium on Circuits and Systems**, v. 2017-August, p. 1260–1263, 2017. ISSN 15483746.

ELSHAFIEY, A. T.; ZARKESH-HA, P.; TRUJILLO, J. The effect of power supply ramp time on SRAM PUFs. **Midwest Symposium on Circuits and Systems**, v. 2017-August, p. 946–949, 2017. ISSN 15483746.

ESCHE, M.; THIEL, F. Software Risk Assessment for Measuring Instruments in Legal Metrology. In: **Proceedings of the Federated Conference on Computer Science and Information Systems**. [s.n.], 2015. v. 5, p. 1113–1123. ISBN 9788360810668. Disponível em: <https://fedcsis.org/proceedings/2015/drp/127.html>.

FOMICHEV, M. et al. Perils of zero-interaction security in the internet of things. **Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies**, ACM New York, NY, USA, v. 3, n. 1, p. 1–38, 2019.

FORTE, D.; BHUNIA, S.; TEHRANIPOOR, M. M. **Hardware protection through obfuscation**. Cham: Springer International Publishing, 2017. 1–349 p. ISBN 9783319490199. Disponível em: <http://link.springer.com/10.1007/978-3-319-49019-9>.

FOURNARIS, A. P.; LAMPROPOULOS, K.; KOUFOPAVLOU, O. Hardware security for critical infrastructures-the cipsec project approach. In: IEEE. **2017 IEEE computer society annual symposium on VLSI (ISVLSI)**. [S.l.], 2017. p. 356–361.

GANGULY, P. et al. Analysis of the security anomalies in the smart metering infrastructure and its impact on energy profiling and measurement. **SMARTGREENS 2016 - Proceedings of the 5th International Conference on Smart Cities and Green ICT Systems**, n. Smartgreens, p. 302–308, 2016.

GHALEB, S. M. et al. Mobility management for IoT: a survey. **Eurasip Journal on Wireless Communications and Networking**, EURASIP Journal on Wireless Communications and Networking, v. 2016, n. 1, 2016. ISSN 16871499. Disponível em: <http://dx.doi.org/10.1186/s13638-016-0659-4>.

GHOSAL, A.; CONTI, M. Key management systems for smart grid advanced metering infrastructure: A survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 21, n. 3, p. 2831–2848, 2019.

GIECHASKIEL, I.; RASMUSSEN, K. Taxonomy and challenges of out-of-band signal injection attacks and defenses. **IEEE Communications Surveys & Tutorials**, IEEE, v. 22, n. 1, p. 645–670, 2019.

GOYAL, R.; DRAGONI, N.; SPOGNARDI, A. Mind the tracker you wear: a security analysis of wearable health trackers. In: **Proceedings of the 31st Annual ACM Symposium on Applied Computing**. [S.l.: s.n.], 2016. p. 131–136.

Griffor Edward R., Greer Christopher, Wollman David A., B. M. J. **Framework for Cyber-Physical Systems : Volume 1 , Overview NIST Special Publication 1500-201 Framework for Cyber-Physical Systems : Volume 1 , Overview**. Gaithersburg, MD, 2017. v. 1, n. 1, 79 p. Disponível em: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-201.pdf>.

GUAJARDO, J. et al. FPGA Intrinsic PUFs and Their Use for IP Protection. In: **Conference on Cryptographic Hardware and Embedded Systems**. Vienna, Austria,: Springer, 2007. p. 63–80.

GUIDE, E. C. L. M. **(WELMEC), Braunschweig, Germany**. 2008. Disponível em: <https://www.welmec.org/>.

GUIN, U.; ASADIZANJANI, N.; TEHRANIPOOR, M. Standards for hardware security. **GetMobile: Mobile Computing and Communications**, ACM New York, NY, USA, v. 23, n. 1, p. 5–9, 2019.

GUIZANI, M. The Industrial Internet of Things. **IEEE Network**, v. 33, n. 5, p. 4–4, sep 2019. ISSN 0890-8044. Disponível em: <https://ieeexplore.ieee.org/document/8863716/>.

GUPTA, R.; GUPTA, P.; SINGH, J. **Security and Cryptography**. Second edition. Elsevier Inc., 2019. 501–547 p. ISBN 9780128094488. Disponível em: <https://doi.org/10.1016/B978-0-12-809448-8.00014-X>.

Günlü, O. et al. Code constructions for physical unclonable functions and biometric secrecy systems. **IEEE Transactions on Information Forensics and Security**, v. 14, n. 11, p. 2848–2858, 2019.

HABIB, K.; LEISTER, W. Context-Aware Authentication for the Internet of Things. **The Eleventh International Conference on Autonomic and Autonomous Systems fined**, n. c, p. 134–139, 2015.

Hallmans, D. et al. A method and industrial case: Replacement of an fpga component in a legacy control system. In: **2015 IEEE 13th International Conference on Industrial Informatics (INDIN)**. [S.l.: s.n.], 2015. p. 208–214.

HANSEN, M.; SCHWARTZ, A.; COOPER, A. Privacy and identity management. **IEEE Security and Privacy**, v. 6, n. 2, p. 38–45, 2008. ISSN 15407993.

HASSIJA, V. et al. A Survey on Supply Chain Security: Application Areas, Security Threats, and Solution Architectures. **IEEE Internet of Things Journal**, v. 333031, n. c, p. 1–1, 2020. ISSN 2327-4662. Disponível em: <https://ieeexplore.ieee.org/document/9203862/>.

HE, J. et al. Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, IEEE, v. 25, n. 10, p. 2939–2948, 2017.

HELFMEIER, C. et al. Cloning physically unclonable functions. **Proceedings of the 2013 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2013**, IEEE, p. 1–6, 2013.

HERDER, C. et al. Physical unclonable functions and applications: A tutorial. **Proceedings of the IEEE**, IEEE, v. 102, n. 8, p. 1126–1141, 2014. ISSN 00189219.

HU, C.-c.; WEI, H.-x.; B, M.-t. C. **Shareflow : A Visualization Tool**. Springer International Publishing, 2019. v. 1. 563–581 p. ISBN 9783030190637. Disponível em: <http://dx.doi.org/10.1007/978-3-030-19063-7_45>.

HU, W. et al. Detecting Hardware Trojans with Gate-Level Information-Flow Tracking. **Computer**, IEEE, v. 49, n. 8, p. 44–52, aug 2016. ISSN 00189162. Disponível em: <http://ieeexplore.ieee.org/document/7543420/>.

HUSSAIN, R.; ZEADALLY, S. Autonomous cars: Research results, issues, and future challenges. **IEEE Communications Surveys & Tutorials**, IEEE, v. 21, n. 2, p. 1275–1313, 2018.

IEHIRA, K.; INOUE, H.; ISHIDA, K. Spoofing attack using bus-off attacks against a specific ecu of the can bus. In: IEEE. **2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)**. [S.l.], 2018. p. 1–4.

INMETRO. **ORIENTAÇÕES PARA REDAÇÃO DE MEMORIAL DESCRITIVO DE SOFTWARE PARA MEDIDORES DE ENERGIA ELÉTRICA**. 2018. Urlhttp://www.inmetro.gov.br/metlegal/docDisponiveis.asp.

IVANOV, R.; WEIMER, J.; LEE, I. Towards context-aware cyber-physical systems. In: IEEE. **2018 IEEE Workshop on Monitoring and Testing of Cyber-Physical Systems (MT-CPS)**. [S.l.], 2018. p. 10–11.

JAYATILLEKA, I.; HALGAMUGE, M. N. **Internet of Things in healthcare: Smart devices, sensors, and systems related to diseases and health conditions**. Elsevier Inc., 2020. 1–35 p. ISBN 9780128180143. Disponível em: <http://dx.doi.org/10.1016/B978-0-12-818014-3.00001-2>.

JELOKA, S. et al. A sequence dependent challenge-response puf using 28nm sram 6t bit cell. In: IEEE. **2017 Symposium on VLSI Circuits**. [S.l.], 2017. p. C270–C271.

JIANG, Q. et al. Shake to communicate: Secure handshake acceleration-based pairing mechanism for wrist worn devices. **IEEE Internet of Things Journal**, IEEE, v. 6, n. 3, p. 5618–5630, 2019. ISSN 23274662.

KABALCI, Y. A survey on smart metering and smart grid communication. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 57, p. 302–318, 2016.

KARAPANOS, N. et al. Sound-proof: Usable two-factor authentication based on ambient sound. **Proceedings of the 24th USENIX Security Symposium**, p. 483–498, 2015.

KAZEMI, Z. et al. Hardware Security Vulnerability Assessment to Identify the Potential Risks in A Critical Embedded Application. In: **Proceedings - 2020 26th IEEE International Symposium on On-Line Testing and Robust System Design, IOLTS 2020**. IEEE, 2020. v. 6, p. 1–6. ISBN 9781728181875. Disponível em: <https://ieeexplore.ieee.org/document/9159739/>.

KHELIF, M. A. et al. Toward a hardware man-in-the-middle attack on PCIe bus. **Microprocessors and Microsystems**, Elsevier B.V., v. 77, 2020. ISSN 01419331.

KIM, K. et al. Cybersecurity for autonomous vehicles: Review of attacks and defense. **Computers & Security**, Elsevier Ltd, v. 103, p. 102150, apr 2021. ISSN 01674048. Disponível em: <https://doi.org/10.1016/j.cose.2020.102150https://linkinghub.elsevier.com/retrieve/pii/S0167404820304235>.

KOCHER, P. et al. Spectre attacks: Exploiting Speculative Execution. **Communications of the ACM**, IEEE, v. 63, n. 7, p. 93–101, 2020. ISSN 15577317.

KONSTANTINOU, C.; KELIRIS, A.; MANIATAKOS, M. Taxonomy of firmware trojans in smart grid devices. In: IEEE. **2016 IEEE Power and Energy Society General Meeting (PESGM)**. [S.l.], 2016. p. 1–5.

KONSTANTINOU, C.; MANIATAKOS, M. Hardware-Layer Intelligence Collection for Smart Grid Embedded Systems. **Journal of Hardware and Systems Security**, Journal of Hardware and Systems Security, v. 3, n. 2, p. 132–146, 2019. ISSN 2509-3428.

KORAK, T.; HOEFLER, M. On the effects of clock and power supply tampering on two microcontroller platforms. **Proceedings - 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014**, IEEE, p. 8–17, 2014.

KOTESHWARA, S.; DAS, A. Comparative Study of Authenticated Encryption Targeting Lightweight IoT Applications. **IEEE Design and Test**, IEEE, v. 34, n. 4, p. 26–33, 2017. ISSN 21682356.

KROEGER, T. et al. Effect of Aging on PUF Modeling Attacks based on Power Side-Channel Observations. **Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition, DATE 2020**, p. 454–459, 2020.

KUMAR, R.; DHANUSKODI, S. N.; KUNDU, S. On Manufacturing Aware Physical Design to Improve the Uniqueness of Silicon-Based Physically Unclonable Functions. IEEE, 2014.

KURINEC, S. K.; INIEWSKI, K. **Nanoscale semiconductor memories: Technology and applications**. [S.l.]: CRC press, 2013.

LEE, R. P.; MARKANTONAKIS, K.; AKRAM, R. N. Binding hardware and software to prevent firmware modification and device counterfeiting. **CPSS 2016 - Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security, Co-located with Asia CCS 2016**, p. 70–81, 2016.

LÉGALE, O. I. de M. **General requirements for software controlled measuring instruments**. 2008. Disponível em: <https://www.oiml.org/en/files/pdf_d/d031-e08.pdf/at_download/file>.

LEITÃO, F. O.; VASCONCELLOS, M. T.; BRANDÃO, P. C. R. Hardware and Software Countermeasures on High Technology Fraud at Fuel Dispensers under the Scope of Legal Metrology. In: **IX Simposio Internacional 'Metrologia 2014'**. Havana: [s.n.], 2014. p. 1–10.

LI, J. et al. K-Hunt. p. 412–425, 2018.

LI, K. F.; ATTARMOGHADDAM, N. Challenges and methodologies of hardware security. In: IEEE. **2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)**. [S.l.], 2018. p. 928–933.

LI, W.; SONG, H.; ZENG, F. Policy-Based Secure and Trustworthy Sensing for Internet of Things in Smart Cities. **IEEE Internet of Things Journal**, IEEE, v. 5, n. 2, p. 716–723, 2018. ISSN 23274662.

LIANG, W. et al. Study on puf based secure protection for ic design. **Microprocessors and Microsystems**, Elsevier, v. 45, p. 56–66, 2016.

LIANG, Z. Y.; WEI, H. H.; LIU, T. T. A Wide-Range Variation-Resilient Physically Unclonable Function in 28 nm. **IEEE Journal of Solid-State Circuits**, IEEE, v. 55, n. 3, p. 817–825, 2020. ISSN 1558173X.

LIAO, B. et al. Security Analysis of IoT Devices by Using Mobile Computing: A Systematic Literature Review. **IEEE Access**, v. 8, p. 120331–120350, 2020. ISSN 21693536.

LIPP, M. et al. Meltdown: Reading Kernel Memory from User Space. **Communications of the ACM**, v. 63, n. 6, p. 46–56, 2020. ISSN 15577317.

LIU, C. et al. A flexible hardware architecture for slave device of I2C bus. **Proceedings - 2019 International Conference on Electronic Engineering and Informatics, EEI 2019**, p. 309–313, 2019.

LIU, H. et al. Smart solution, poor protection: An empirical study of security and privacy issues in developing and deploying smart home devices. In: **Proceedings of the 2017 Workshop on Internet of Things Security and Privacy**. [S.l.: s.n.], 2017. p. 13–18.

LOZOYA-SANTOS, J. d. J. et al. Survey on biometry for cognitive automotive systems. **Cognitive Systems Research**, v. 55, p. 175–191, 2019. ISSN 13890417.

MAES, R.; Van Der Leest, V. Countering the effects of silicon aging on SRAM PUFs. **Proceedings of the 2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014**, p. 148–153, 2014.

MAES, R.; VERBAUWHEDE, I. Physically unclonable functions: A study on the state of the art and future research directions. In: **Towards Hardware-Intrinsic Security**. [S.l.]: Springer, 2010. p. 3–37.

MALDINI, A. et al. Optimizing Electromagnetic Fault Injection with Genetic Algorithms. **Automated Methods in Cryptographic Fault Analysis**, p. 281–300, 2019.

MARTIN, H. et al. A New TRNG Based on Coherent Sampling with Self-Timed Rings. **IEEE Transactions on Industrial Informatics**, IEEE, v. 12, n. 1, p. 91–100, 2016. ISSN 15513203.

MAYRHOFER, R.; GELLERSEN, H. Shake well before use: Authentication based on accelerometer data. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 4480 LNCS, p. 144–161, 2007. ISSN 16113349.

MEIJER, C.; MOONSAMY, V.; WETZELS, J. Where's Crypto?: Automated Identification and Classification of Proprietary Cryptographic Primitives in Binary Code. 2020. Disponível em: <http://arxiv.org/abs/2009.04274>.

Melo Jr, W. et al. Certificação Digital como Ferramenta de Segurança para Medidores Inteligentes. In: **Anais Estendidos do Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC)**. Sociedade Brasileira de Computação - SBC, 2019. p. 89–94. Disponível em: <https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/8641>.

Melo Jr, W. et al. Public-Key Infrastructure for Smart Meters using Blockchains. In: **2020 IEEE International Workshop on Metrology for Industry 4.0 and IoT**. [S.l.: s.n.], 2020. p. 6.

Melo Jr., W. S. et al. Using Blockchains to Implement Distributed Measuring Systems. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 68, n. 5, p. 1503–1512, 2019. ISSN 0018-9456. Disponível em: <https://ieeexplore.ieee.org/document/8667365/>.

MELO, W. et al. How blockchains can improve measuring instruments regulation and control. **I2MTC 2018 - 2018 IEEE International Instrumentation and Measurement Technology Conference: Discovering New Horizons in Instrumentation and Measurement, Proceedings**, IEEE, p. 1–6, 2018.

MELO, W. et al. Public-Key Infrastructure for Smart Meters using Blockchains. **2020 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2020 - Proceedings**, p. 429–434, 2020.

MELO, W. S. **Autenticação baseada em Contexto Físico: Metodologia, Protocolos e Aplicações**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, Tese (Doutorado em Informática), 2018.

MELO, W. S.; MACHADO, R.; CARMO, L. F. Using physical context-based authentication against external attacks: Models and protocols. **Security and Communication Networks**, Hindawi, v. 2018, 2018.

MIAO, J. et al. Lrr-dpuf: Learning resilient and reliable digital physical unclonable function. In: IEEE. **2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)**. [S.l.], 2016. p. 1–8.

MICHAEL, H. Investigating the Vulnerabilities of Two Microcontroller Platforms to Fault Injection Attacks. 2014.

MILLER, C. Lessons learned from hacking a car. **IEEE Design & Test**, IEEE, v. 36, n. 6, p. 7–9, 2019.

MISHRA, P.; BHUNIA, S.; TEHRANIPOOR, M. **Hardware IP security and trust**. [S.l.]: Springer, 2017.

MITROPOULOS, D. et al. How to train your browser: Preventing XSS attacks using contextual script fingerprints. **ACM Transactions on Privacy and Security**, v. 19, n. 1, p. 1–31, aug 2016. ISSN 24712574. Disponível em: <https://dl.acm.org/doi/10.1145/2939374>.

MOIS, G.; SANISLAV, T.; FOLEA, S. C. A Cyber-Physical System for Environmental Monitoring. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 65, n. 6, p. 1463–1471, 2016. ISSN 00189456.

MOLLAH, M. B.; AZAD, M. A. K.; VASILAKOS, A. Security and privacy challenges in mobile cloud computing: Survey and way ahead. **Journal of Network and Computer Applications**, Elsevier, v. 84, p. 38–54, 2017.

MUHAL, M. A. et al. Physical unclonable function based authentication scheme for smart devices in internet of things. In: IEEE. **2018 IEEE International Conference on Smart Internet of Things (SmartIoT)**. [S.l.], 2018. p. 160–165.

MULLEN, G.; MEANY, L. Assessment of buffer overflow based attacks on an IoT operating system. **Global IoT Summit, GIoTS 2019 - Proceedings**, IEEE, p. 1–6, 2019.

MUSHTAQ, M. F. et al. A survey on the cryptographic encryption algorithms. **International Journal of Advanced Computer Science and Applications**, v. 8, n. 11, p. 333–344, 2017.

NAGRA, J.; COLLBERG, C. **Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection**. [S.l.]: Pearson Education, 2009.

NEDOSPASOV, D. et al. Invasive PUF Analysis. In: **2013 Workshop on Fault Diagnosis and Tolerance in Cryptography**. IEEE, 2013. p. 30–38. ISBN 978-0-7695-5059-6. Disponível em: <http://ieeexplore.ieee.org/document/6623553/>.

OKTAVIA, T. et al. Security and privacy challenge in bring your own device environment: A systematic literature review. In: IEEE. **2016 International Conference on Information Management and Technology (ICIMTech)**. [S.l.], 2016. p. 194–199.

PALEY, S.; HOQUE, T.; BHUNIA, S. Active protection against pcb physical tampering. In: IEEE. **2016 17th International Symposium on Quality Electronic Design (ISQED)**. [S.l.], 2016. p. 356–361.

PAPPU, R. et al. Physical one-way functions. **Science**, v. 297, n. 5589, p. 2026–2030, 2002. ISSN 00368075.

PAUL, S. et al. A review of smart technology (smart grid) and its features. In: **Proceedings of 2014 1st International Conference on Non Conventional Energy: Search for Clean and Safe Energy, ICONCE 2014**. IEEE, 2014. p. 200–203. ISBN 978-1-4799-3340-2. Disponível em: <http://ieeexplore.ieee.org/document/6808719/>.

PEREIRA, L. et al. Using intel SGX to enforce auditing of running software in insecure environments. **Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom**, v. 2018-Decem, p. 243–246, 2018. ISSN 23302186.

PETERS, D. et al. IT Security for Measuring Instruments : Confidential Checking of Software Functionality In book : Advances in Information and Communication Conference : Future of Information and Communication Conference (FICC) 2020 , 5-6 March , San Francisco IT Securi. n. February, 2020.

PETERS, D. et al. A secure software framework for measuring instruments in legal metrology. In: IEEE. **2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings**. [S.l.], 2015. p. 1596–1601.

PINTO, S. et al. Lightweight multicore virtualization architecture exploiting ARM TrustZone. **Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society**, v. 2017-January, p. 3562–3567, 2017.

PRADO, C. B. do et al. Software Analysis and Protection for Smart Metering. **NCSLI Measure**, v. 9, n. 3, p. 22–29, 2014. ISSN 1931-5775.

PRINETTO, P.; ROASCIO, G. Hardware security, vulnerabilities, and attacks: A comprehensive taxonomy. In: **ITASEC**. [S.l.: s.n.], 2020. p. 177–189.

PROKOFIEV, A. O.; SMIRNOVA, Y. S.; SILNOV, D. S. The Internet of Things cybersecurity examination. In: **Proceedings - 2017 Siberian Symposium on Data Science and Engineering, SSDSE 2017**. IEEE, 2017. p. 44–48. ISBN 9781538615935. Disponível em: <http://ieeexplore.ieee.org/document/8071962/>.

QIN, Y. et al. RIPTE: Runtime Integrity Protection Based on Trusted Execution for IoT Device. **Security and Communication Networks**, v. 2020, 2020. ISSN 19390122.

QU, Y. J. et al. Smart manufacturing systems: state of the art and future trends. **The International Journal of Advanced Manufacturing Technology**, v. 103, n. 9-12, p. 3751–3768, aug 2019. ISSN 0268-3768. Disponível em: <http://link.springer.com/10.1007/s00170-019-03754-7>.

RAGAB, A. et al. Robust Hybrid Lightweight Cryptosystem for Protecting IoT Smart Devices. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. Springer International Publishing, 2019. v. 11637 LNCS, p. 5–19. ISBN 9783030248994. Disponível em: <http://dx.doi.org/10.1007/978-3-030-24900-7_1http://link.springer.com/10.1007/978-3-030-24900-7_1>.

REVERTER, F.; GASULLA, M. Experimental study on the power consumption of timers embedded into microcontrollers. IEEE, p. 1–5, 2021.

Rodrigues Filho, B. A.; GONÇALVES, R. F. Legal metrology, the economy and society: A systematic literature review. **Measurement**, Elsevier Ltd, v. 69, p. 155–163, 2015. ISSN 02632241.

ROSTAMI, M.; JUELS, A.; KOUSHANFAR, F. Heart-to-heart (H2H). In: **Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13**. New York, New York, USA: ACM Press, 2013. p. 1099–1112. ISBN 9781450324779. Disponível em: <http://dl.acm.org/citation.cfm?doid=2508859.2516658>.

RÜHRMAIR, U.; HOLCOMB, D. E. PUFs at a glance. **Proceedings -Design, Automation and Test in Europe, DATE**, 2014. ISSN 15301591.

RÜHRMAIR, U.; SCHLICHTMANN, U.; BURLESON, W. Special session: How secure are PUFs really? on the reach and limits of recent PUF attacks. **Proceedings -Design, Automation and Test in Europe, DATE**, p. 12–15, 2014. ISSN 15301591.

RUI, Z.; YAN, Z. A survey on biometric authentication: Toward secure and privacy-preserving identification. **IEEE Access**, IEEE, v. 7, p. 5994–6009, 2018.

RUKHIN, A.; SOTO, J.; NECHVATAL, J. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. **Nist Special Publication**, v. 22, n. April, p. 1/1—-G/1, 2010. Disponível em: <http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22rev1a.pdf>.

SAKHARE, S.; SAKHARE, D. A review—hardware security using puf (physical unclonable function). In: **ICCCE 2019**. [S.l.]: Springer, 2020. p. 373–377.

SALLAM, S.; BEHESHTI, B. D. A Survey on Lightweight Cryptographic Algorithms. **IEEE Region 10 Annual International Conference, Proceedings/TENCON**, IEEE, v. 2018-October, n. October, p. 1784–1789, 2019. ISSN 21593450.

SANCHEZ, I.; SANTOS, I.; BALZAROTTI, D. Clock around the clock: Time-based device fingerprinting. **Proceedings of the ACM Conference on Computer and Communications Security**, p. 1502–1514, 2018. ISSN 15437221.

SANCHEZ, P. M. S. et al. A Survey on Device Behavior Fingerprinting: Data Sources, Techniques, Application Scenarios, and Datasets. **IEEE Communications Surveys and Tutorials**, v. 23, n. 2, p. 1048–1077, 2021. ISSN 1553877X.

SANDHU, R. et al. Identification and authentication. **Computer Security Handbook**, Wiley Online Library, p. 28–1, 2012.

SARRAB, M.; ALNAELI, S. M. Critical Aspects Pertaining Security of IoT Application Level Software Systems. **2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018**, IEEE, p. 960–964, 2019.

SAU, S. et al. Survey of secure processors. In: IEEE. **2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)**. [S.l.], 2017. p. 253–260.

SERPANOS, D. N.; VOYIATZIS, A. G. Security challenges in embedded systems. **ACM Transactions on embedded computing systems (TECS)**, ACM New York, NY, USA, v. 12, n. 1s, p. 1–10, 2013.

SHI, J. et al. A survey of Cyber-Physical Systems. In: **2011 International Conference on Wireless Communications and Signal Processing, WCSP 2011**. IEEE, 2011. p. 1–6. ISBN 9781457710100. Disponível em: <http://ieeexplore.ieee.org/document/6096958/>.

SHIN, C. et al. **Variation-aware advanced CMOS devices and SRAM**. [S.l.]: Springer, 2016. v. 56.

SHOUKRY, Y. et al. Non-invasive spoofing attacks for anti-lock braking systems. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 8086 LNCS, p. 55–72, 2013. ISSN 16113349.

SHROUF, F.; ORDIERES, J.; MIRAGLIOTTA, G. Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the internet of things paradigm. In: IEEE. **2014 IEEE international conference on industrial engineering and engineering management**. [S.l.], 2014. p. 697–701.

SHU, X.; YAO, D.; BERTINO, E. Privacy-preserving detection of sensitive data exposure. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 10, n. 5, p. 1092–1103, 2015. ISSN 15566013.

SHWARTZ, O. et al. Inner conflict: How smart device components can cause harm. **Computers & Security**, Elsevier, v. 89, p. 101665, 2020.

SHWARTZ, O. et al. From smashed screens to smashed stacks: attacking mobile phones using malicious aftermarket parts. In: IEEE. **2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)**. [S.l.], 2017. p. 94–98.

SIDDIQUI, A. S. et al. Hardware assisted security architecture for smart grid. **Proceedings: IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society**, IEEE, p. 2890–2895, 2018.

SIDERIS, A. et al. Smart grid hardware security. In: **IoT for Smart Grids**. [S.l.]: Springer, 2019. p. 85–113.

SLEIT, A.; FETAIS, N. Watermarking: A review of software and hardware techniques. In: IEEE. **2018 International Conference on Computational Science and Computational Intelligence (CSCI)**. [S.l.], 2018. p. 397–403.

SONAQUBE. **Sonaqube**. 2021. Urlhttps://www.sonarqube.org/.

SONG, B. et al. Environmental-Variation-Tolerant magnetic tunnel junction-based physical unclonable function cell with auto write-back technique. **IEEE Transactions on Information Forensics and Security**, v. 16, p. 2843–2853, 2021. ISSN 15566021.

SRIDHAR, S.; SMYS, S. Intelligent security framework for iot devices cryptography based end-to-end security architecture. In: IEEE. **2017 International Conference on Inventive Systems and Control (ICISC)**. [S.l.], 2017. p. 1–5.

STALLINGS, W. **Computer Security Princuplew and Practices**. [S.l.: s.n.], 2014. 200,201 p. ISBN 9780133773927.

STANCIU, A.; MOLDOVEANU, F. D.; CIRSTEA, M. A novel PUF-based encryption protocol for embedded System on Chip. **2016 13th International Conference on Development and Application Systems, DAS 2016 - Conference Proceedings**, IEEE, n. 11, p. 158–165, 2016.

SURI, M.; CHAKRABORTY, S. High-quality puf extraction from commercial rram using switching-time variability. In: IEEE. **2018 IEEE International Memory Workshop (IMW)**. [S.l.], 2018. p. 1–4.

SUTAR, S.; RAHA, A.; RAGHUNATHAN, V. D-puf: An intrinsically reconfigurable dram puf for device authentication in embedded systems. In: IEEE. **2016 International Conference on Compliers, Architectures, and Sythesis of Embedded Systems (CASES)**. [S.l.], 2016. p. 1–10.

TANHA, M. et al. An overview of attacks against digital watermarking and their respective countermeasures. In: IEEE. **Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)**. [S.l.], 2012. p. 265–270.

Tanjidur Rahman, M. et al. Physical inspection attacks: New frontier in hardware security. **2018 IEEE 3rd International Verification and Security Workshop, IVSW 2018**, IEEE, p. 93–102, 2018.

TEHRANIPOOR, F. et al. DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, IEEE, v. 25, n. 3, p. 1085–1097, 2017. ISSN 10638210.

TEHRANIPOOR, F. et al. DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 25, n. 3, p. 1085–1097, 2017. ISSN 10638210.

TEHRANIPOOR, M.; KOUSHANFAR, F. A survey of hardware trojan taxonomy and detection. **IEEE Design and Test of Computers**, v. 27, n. 1, p. 10–25, 2010. ISSN 07407475.

TENORIO, V. et al. Low-Cost, Practical Data Confidentiality Support for IoT Data Sources. **Brazilian Symposium on Computing System Engineering, SBESC**, v. 2019-November, 2019. ISSN 23247894.

TOMIĆ, I.; MCCANN, J. A. A survey of potential security issues in existing wireless sensor network protocols. **IEEE Internet of Things Journal**, IEEE, v. 4, n. 6, p. 1910–1923, 2017.

TSOUTSOS, N. G.; KONSTANTINOU, C.; MANIATAKOS, M. Advanced techniques for designing stealthy hardware trojans. In: IEEE. **2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)**. [S.l.], 2014. p. 1–4.

TUYLS, P. **Towards hardware-intrinsic security: foundations and practice**. [S.l.]: Springer Science & Business Media, 2010.

VALEA, E. et al. A survey on security threats and countermeasures in ieee test standards. **IEEE Design & Test**, IEEE, v. 36, n. 3, p. 95–116, 2019.

VEERANNA, N.; SCHAFER, B. C. Hardware trojan detection in behavioral intellectual properties (ip's) using property checking techniques. **IEEE Transactions on Emerging Topics in Computing**, IEEE, v. 5, n. 4, p. 576–585, 2016.

VERACODE. **Veracode**. 2021. Urlhttps://www.veracode.com/.

VIJAYAKUMAR, A.; PATIL, V. C.; KUNDU, S. On Testing Physically Unclonable Functions for Uniqueness. IEEE, 2016.

WACHSMANN, C.; SADEGHI, A.-R. Physically unclonable functions (pufs): Applications, models, and future directions. **Synthesis Lectures on Information Security, Privacy, & Trust**, Morgan & Claypool Publishers, v. 5, n. 3, p. 1–91, 2014.

WANG, R. et al. Long-term Continuous Assessment of SRAM PUF and Source of Random Numbers. **Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition, DATE 2020**, p. 7–12, 2020.

WANG, S. et al. Security in wearable communications. **IEEE Network**, IEEE, v. 30, n. 5, p. 61–67, 2016.

WANG, S. et al. Security in wearable communications. **IEEE Network**, IEEE, v. 30, n. 5, p. 61–67, 2016. ISSN 08908044.

WANG, W. et al. Securing On-Body IoT Devices by Exploiting Creeping Wave Propagation. **IEEE Journal on Selected Areas in Communications**, v. 36, n. 4, p. 696–703, 2018. ISSN 07338716.

WANG, X.; TEHRANIPOOR, M. Novel physical unclonable function with process and environmental variations. **Proceedings -Design, Automation and Test in Europe, DATE**, IEEE, p. 1065–1070, 2010. ISSN 15301591.

WANG, Y. et al. Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. **IEEE Transactions on Smart Grid**, IEEE, v. 10, n. 3, p. 3125–3148, may 2019. ISSN 1949-3053. Disponível em: <https://ieeexplore.ieee.org/document/8322199/>.

WARUDKAR, D.; CHANDEL, P.; SAWALE, B. A. Anti-Tamper Features in Electronic Energy Meters. **International Journal of Electrical, Electronics and Data Communication**, n. 2, p. 2320–2084, 2014.

WERANGA, K. S.; KUMARAWADU, S.; CHANDIMA, D. P. **Smart metering design and applications**. Singapore: Springer Singapore, 2014. (SpringerBriefs in Applied Sciences and Technology, 9789814451819). ISSN 21915318. ISBN 978-981-4451-81-9. Disponível em: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/9814451819http://link.springer.com/10.1007/978-981-4451-82-6>.

WILLIAMS, R. et al. Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach. **2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data, ISI 2017**, p. 179–181, 2017.

WURM, J. et al. Security analysis on consumer and industrial IoT devices. **Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC**, v. 25-28-Janu, p. 519–524, 2016.

XU, B. et al. Ubiquitous data accessing method in iot-based information system for emergency medical services. **IEEE Transactions on Industrial informatics**, IEEE, v. 10, n. 2, p. 1578–1586, 2014.

XU, T.; WENDT, J. B.; POTKONJAK, M. Security of iot systems: Design challenges and opportunities. In: IEEE. **2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)**. [S.l.], 2014. p. 417–423.

XUE, M. et al. Ten years of hardware trojans: a survey from the attacker's perspective. **IET Computers & Digital Techniques**, IET, v. 14, n. 6, p. 231–246, 2020.

YACCHIREMA, D. C.; ESTEVE, M.; PALAU, C. E. Design and implementation of a gateway for pervasive smart environments. In: IEEE. **2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.], 2016. p. 004454–004459.

YAGER, N.; DUNSTONE, T. The biometric menagerie. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 32, n. 2, p. 220–230, feb 2010. ISSN 01628828. Disponível em: <http://ieeexplore.ieee.org/document/4711054/>.

YU, W. et al. A study of HSM based key protection in encryption file system. **2016 IEEE Conference on Communications and Network Security, CNS 2016**, IEEE, p. 352–353, 2017.

ZEITOUNI, S. et al. Remanence Decay Side-Channel: The PUF Case. **IEEE Transactions on Information Forensics and Security**, IEEE, v. 11, n. 6, p. 1106–1116, 2016. ISSN 15566013.

ZENGER, C. T. et al. Exploiting the physical environment for securing the internet of things. In: **Proceedings of the 2015 New Security Paradigms Workshop**. [S.l.: s.n.], 2015. p. 44–58.

ZHANG, M.; SEKAR, R. Control Flow and Code Integrity for COTS binaries. **Proceedings of the 31st Annual Computer Security Applications Conference on - ACSAC 2015**, p. 91–100, 2015. Disponível em: <http://dl.acm.org/citation.cfm?id=2818000.2818016%5Cnhttp://dl.acm.org/citation.cfm?doid=2818000.2818016>.

ZHANG, S.; ZHENG, T.; WANG, B. A privacy protection scheme for smart meter that can verify terminal's trustworthiness. **International Journal of Electrical Power and Energy Systems**, Elsevier, v. 108, n. November 2018, p. 117–124, 2019. ISSN 01420615. Disponível em: <https://doi.org/10.1016/j.ijepes.2019.01.010>.

ZHANG, X.; WANG, P.; ZHANG, Y. Highly stable data SRAM-PUF in 65nm CMOS process. **Proceedings of International Conference on ASIC**, p. 0–3, 2013. ISSN 2162755X.

ZHANG, Z. et al. Securing fpga-based obsolete component replacement for legacy systems. In: IEEE. **2018 19th International Symposium on Quality Electronic Design (ISQED)**. [S.l.], 2018. p. 401–406.