UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA

Autenticação para Circuitos Integrados e Dispositivos usando Blockchain e Funções Físicas Não-Clonáveis

Alessandro Augusto Nunes Campos

Maio de 2022
Itajubá M.G.

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA

Alessandro Augusto Nunes Campos

Autenticação para Circuitos Integrados e Dispositivos usando Blockchain e Funções Físicas Não-Clonáveis

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Doutor em Ciências em Engenharia Elétrica

Área de concentração: Microeletrônica

Orientador: Prof. Dr. Tales Cleber Pimenta

Maio de 2022
Itajubá M.G.

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA

Alessandro Augusto Nunes Campos

Autenticação para Circuitos Integrados e Dispositivos usando Blockchain e Funções Físicas Não-Clonáveis

Tese avaliada por banca em 27 de maio de 2022.

Banca examinadora:

Prof. Dr. Tales Cleber Pimenta, UNIFEI (Orientador)
Prof. Dr. Robson Luiz Moreno, UNIFEI
Prof. Dr. Gabriel Antonio Fanelli de Souza, UNIFEI
Prof. Dr. Dalton Martini Colombo, UFMG
Dr. Hamilton José Mendes da Silva, MCTI

Itajubá M.G.
2022

Dedicada à minha família como um todo, em especial a minha esposa Daniela pelos dias e horas de minha ausência e aos meus filhos João Matheus e Isabella, onde espero que o trabalho científico e a busca pelo conhecimento um dia possam inspirá-los.

Agradecimentos

Como não se pode esquecer, agradeço a Deus por me permitir caminhar neste percurso tão desafiador, ainda mais nestes últimos tempos tão difíceis de pandemia, onde até então a referência de normal era outra. E a minha família, pelo apoio, em especial nos momentos mais difíceis de frustação e cansaço.

Agradecimento especial aos professores do programa de pós-graduação em engenharia elétrica da Universidade Federal de Itajubá (UNIFEI), em especial ao professor Dr. Tales Cleber Pimenta pela orientação e pela dedicação em cobrar, ensinar e ajudar nos momentos mais complicados. Agradecer também a tantos outros professores, que sempre se colocaram a disposição em ajudar com os grandes desafios enfrentados neste percurso.

Aos colegas de trabalho e a todos da Secretária de Empreendedorismo e Inovação – SEMPI, do Ministério de Ciência, Tecnologia e Inovações – MCTI, que sempre apoiaram este meu desafio e sempre me incentivaram para a conquista deste objetivo.

Por fim e não menos importante à Universidade Federal de Itajubá – UNFEI, que proporcionou os ambientes, estruturas e as condições necessárias para o avanço e conclusão deste trabalho.

Resumo

Componentes e dispositivos seguros sempre foram e sempre serão um desafio para a indústria eletrônica. Nesse sentido, há uma demanda constante e crescente por novas soluções que possam permitir confiabilidade no uso e autenticidade de componentes e dispositivos. O usuário final não é capaz de avaliar o risco existente, muito menos se o componente ou dispositivo é confiável em vários aspectos, principalmente no acesso indevido às suas informações. Este trabalho apresenta uma nova abordagem com a integração de duas tecnologias: Redes Blockchains, que implementam uma espécie de banco de dados descentralizado e inviolável, que pode aumentar a resiliência, segurança e garantia contra a alteração das informações cadastradas em sua estrutura; Funções Físicas Não Clonáveis (PUF), que permitem a geração de uma chave criptográfica forte, pois utiliza características físicas únicas de cada componente semicondutor, aumentando consideravelmente a segurança, simplicidade, proteção da propriedade industrial e a possibilidade de autenticação remota dos dispositivos. A contribuição aqui está na integração de tecnologias existentes, a fim de obter uma solução inovadora de autenticação e segurança cibernética para a internet das coisas e outros dispositivos.

Palavras-Chave: Blockchain; Circuito integrado; Internet das Coisas; Funções Físicas não Clonáveis; Segurança Cibernética; Proteção de Propriedade Industrial; Autenticação.

Abstract

Secure components and devices have always been and always will be a challenge for the electronics industry. In this sense, there is a constant and growing demand for new solutions that can allow reliability in the use and authenticity of components and devices. The end-user is not able to assess the existing risk, much less if the component or device is reliable in several aspects, mainly improper access to its information. This work presents a new approach with the integration of two tech-nologies: Blockchains networks, which implement a kind of decentralized and inviolable database, which can increase resilience, security and guarantee against the alteration of the information registered in its structure; Physical Unclonable Functions (PUF), which allow the generation of a strong cryptographic key, since they use unique physical characteristics of each semiconductor component, considerably increasing security, simplicity, protection of industrial property and the opportunity for remote authentication of devices. The contribution here is in the integration of existing technologies, in order to obtain an innovative solution of authentication and cyber security for the internet of things and others devices.

KeyWords – BlockChain; Integrated Circuit; Internet of Things; Phisical Unclonable Funcitions; Cyber Security; Industrial Property Protection; Authentication.

Lista de Ilustrações

Lista de tabelas

Lista de símbolos

PUF	Funções Físicas Não Clonáveis *(Physical Unclonable Functions)*
CI	Circuito Integrado
TPM	Módulos de Plataformas Confiáveis (*Trusted Platform Module)*
HSM	Módulos de Hardware Seguros (*Hardware Security Modules)*
ASICs	Circuito Integrado de Aplicação Específica *(Application Specific Integrated Circuit)*
SoC	Sistemas em um chips (*System on a Chip*)
IoT	Internet das Coisas (*Internet of Things)*
SC	Cartões Inteligentes (*Smartcards)*
OTP/ROM	Memória não Volátil Programável Uma Vez (*One Time Programmable/ Read Only Memory*)
EEPROM	Memória Somente de Leitura Programável Apagável Eletricamente (*Electrically Erasable Programmable Read-Only Memory)*
Flash	Tipo de EEPROM
EAL4	Nível de garantia de avaliação do "Critério Comum"
AVA_VAN 5	Análise de vulnerabilidade metódica avançada - Grau 5
IP	Propriedade Intelectual (*Intellectual Properties*)
PCB	Placas de Circuito Impresso *(Printed Circuit Board)*
FRR	Taxa de Rejeição Falsa
FAR	Taxa de Aceitação Falsa
VHDL	Linguagem de Descrição de Hardware *(Hardware Description Language)*
CPU	Unidade Central de Processamento (*Central Process Unit*)
CCD	Dispositivo de carga acoplada *(Charge Coupled Device)*
ICID	Circuito Integrado de Identificação (*Integrated Circuit Identification)*
VT	Tensão de *Threshold*
CMOS	Semicondutor de Óxido Metálico Complementar *(Complementary metal-oxide-semiconductor)*
MOSFET	transistor de efeito de campo de semicondutor de óxido metálico (Metal Oxide Semiconductor Field Effect Transistor)
SRAM	Memória Estática de Acesso Aleatório *(Static Random Access Memory)*

P2P	Ponto a Ponto *(Peer-to-peer)*.
DeFi	Finanças Descentralizadas (*Decentralized Finance*)
ICO	Oferta Inicial de Moedas (*Initial Coin Offering*)
MITM	Ataque por Interceptação de conexão *(Man-in-the-Middle)*
IP	Protocolo de Internet *(Internet Protocol)*
JTAG	Interface de teste em hardware (*Joint Test Action Group)*
USB	Interface de entrada e saída de computador *(Universal Serial Bus)*

Sumário

1 Introdução

É apresentada neste trabalho a proposta de um de sistema de autenticação de segurança de componentes semicondutores (silício) utilizando tecnologias de funções físicas não clonáveis (*Physical Unclonable Functions* – PUFs) e *blockchain* (redes ponto a ponto confiáveis). A principio serão apresentados neste capítulo as considerações necessárias ao entendimento da abordagem proposta, uma descrição do problema abordado e a sua relevância. Será vista também uma solução incluindo seus objetivos, metodologia de pesquisa e contribuições, e a estrutura do trabalho para o avanço e desenvolvimento do estado da arte em segurança cibernética.

1.1 Visão geral

A população de um modo geral pensa que os circuitos integrados (CIs) estão livres dos principais tipos de falhas de segurança tão comuns em software e impermeáveis à subversão de códigos maliciosos [1]. Esta crença é sustentada no caso de módulos de plataformas confiáveis (*Trusted Platform Module* - TPMs) e por Módulos de Hardware Seguros (*Hardware Security Modules* - HSMs), que são dispositivos concebidos com plataformas de alta segurança para processos críticos [2]. Mas essa fé é realmente merecida? Nos últimos tempos, a cadeia de suprimentos foi inundada com componentes falsificados e novos *trojans* de hardware sugiram como ameaça à confiabilidade dos CIs [3].

1.2 Descrição do problema

Usuários finais em geral, não se atém se o hardware utilizado em seus dispositivos é isento de riscos de segurança, crendo que os componentes internos executam suas funções de forma confiável [4][5]. Considerando a possibilidade de realização de acessos indevidos criados ou não deliberadamente por algum fabricante/desenvolvedor ou por falhas em seu desenvolvimento, além de riscos encontrados em circuitos integrados de baixa confiabilidade, este trabalho pretende convergir em um solução que incrementa o uso de CIs de forma segura e com alto grau de inviolabilidade [6][7].

A relevância deste trabalho, se dá especialmente pelo aumento considerável do uso de dispositivos eletrônicos e seus acessórios na vida das pessoas, sendo que, muitas vezes, nos dias atuais, as mesmas confiam suas vidas a estes dispositivos [8].

1.3 Hipótese

Garantia de segurança em hardware não é novidade, mas o tratamento desta, no nível de informações paramétricas, sim. Uma característica pouco explorada é a incapacidade de clonagem de funções físicas (*physical unclonable functions – PUF's*) em silício [9]. O questionamento neste trabalho é como poderiamos desenvolver um solução que utilize estas características para implementar chaves de criptografia para autenticação e/ou habilitação de CIs e outras tecnologias para produzir um nova solução de segurança.

Sabendo-se que apenas a criptografia implementada localmente não se faz, nos dias de hoje, suficientemente forte para resistir as tentativas de uso indevido, adulteração e cópia, necessitamos de soluções inovadoras que melhorem estas características.

Desta forma, temos as cadeias de blocos (*blockchains*), que nada mais são do que um processo de autenticação por criptografia distribuída em rede (*internet*), sendo muito resistentes e resilientes a tais currupções, complementanto a resposta a questão, como uma solução inovadora.

Assim, como hipótese temos: é possível combinar PUFs em silício com a tecnologia blockchain em uma arquitetura mais robusta de autenticação com menor custo, tempo e uso de recursos.

1.4 Objetivos

O objetivo principal deste trabalho é a pesquisa e desenvolvimento de uma solução integrada que, através de implementações específicas, poderá controlar a habilitação, fazer autenticação, além de outras funções em um CI e o protegerão contra uso, cópia e outras violações indevidas, além de garantir sua autenticidade e certificação para o uso.

Conforme abordado na proposição de hipótese deste trabalho, antes do atingimento de suas principais metas tivemos a necessidade de obter conhecimentos das atuais implementações das tecnologias PUFs e *blockchain* e quais as suas vantagens e limitações.

Assim, se fez necessário adquirir informações complementares como a análise exaustiva das tecnologias, seu grau de acessibilidade e implementação, juntamente com os desafios de sua integração.

Por fim, com o resultado obtido pela domínio destes objetivos irtermediários, atingimos ao final uma solução efetiva e comprovadamente relevante e inovadora.

1.5 Metodologia

Durante o desenvolvimento do tema abordado neste trabalho, uma extensa pesquisa bibliográfica foi realizada para revisar as técnicas convencionais de segurança cibernética aplicáveis atualmente em componentes semicondutores, além da busca por trabalhos que pudessem contribuir, mesmo que parcialmente, com o tema. Neste sentido obteve-se um bom material de apoio e consolidou-se a ideia do ineditismo desta proposta.

A partir de então se iniciou a fase de avaliação de um provável modelo, analisando dentro de cada uma das soluções tecnológicas, quais características seriam necessárias para o êxito deste trabalho.

Nesta etapa da metodologia, primeiramente pensando nas soluções de funções físicas não clonáveis, procurou-se avaliar qual solução existente poderia ser utilizada nesta proposta, sua aplicabilidade, facilidade para implementação e comprovação da eficiência, assim com uma pesquisa quantitativa e qualitativa sobre o seu uso. Já na definição da *blockchain*, optou-se por uma solução que pudesse de forma sólida, ser utilizada comercialmente e fizesse uso da melhor opção tecnológica, avaliação de desempenho e segurança. Para isto apresentamos os critérios de avalição e como foi realizado a classificação e seleção da solução. Todas estas etapas da metodologia estão apresentadas no decorrer deste trabalho.

Após isto, consolidada as fases anteriores, estruturou-se o modelo proposto, trabalhando com os critérios já citados, utilizando-se de soluções já desenvolvidas, junto com contribuições específicas criadas para este trabalho pensadas no decorrer e evolução desta proposta.

Por fim, como ultima etapa desta metodologia, avaliamos a solução proposta com relação a ganhos e perdas referentes a outras soluções de mercado do ponto de vista macro, pensado na facilidade de implementação e simplicidade de uso. E finalmente obtivemos pela extração de outras informações na modelagem final, todas considerações e resultados, concluindo assim a metodologia empregada neste trabalho.

1.6 Contribuições

A contribuição efetiva neste trabalho foi a criação de novas técnicas de software e o uso estruturas específicas de arquiteturas de circuitos para aplicação em em segmentos estratégicos, cuja natureza incentiva a utilização de componentes protegidos contra a intrusão ou o acesso não autorizado (defesa, aeroespacial, segredo industrial) ou em aplicações de alto valor agregado.

Os sistemas e subsistemas desenvolvidos aqui, poderão suscitar novos temas de pesquisa indicando outros pontos a serem investigados dos quais poderão ser desenvolvidos em futuros trabalhos de mestrado e/ou doutorado.

Em termos produtivos, os circuitos apresentados podem ser empregados em projetos completos de componentes que necessitem de habilitação certificada e segura e/ou equipamentos que usam componentes de alto valor agregado da qual se deseja proteger de uso impróprio e quebra de propriedade intelectual [10].

1.7 Estrutura do trabalho

O presente trabalho está dividido em quatro capítulos. Neste primeiro capítulo são apresentados, de forma geral, os assuntos abordados durante o desenvolvimento da proposta desta tese. São detalhados também os problemas encontrados na literatura relacionados à segurança cibernética em componentes semicondutores e, além disso, formula-se a hipótese, discorre-se acerca da metodologia, bem como quanto aos objetivos, as contribuições e a estrutura do trabalho.

No segundo capítulo está apresentada toda a teoria e aplicação das funções físicas não clonáveis *(("Physical Unclonable Functions"* – PUFs). Temos neste capítulo a origem das PUFs e as características únicas que as tornam individualizadas como também as principais aplicações onde esta unicidade é relevante. Tratamos neste ponto a caraterização de uma PUF e o que a torna diferente, além de exemplificar outros usos de PUFs que não apenas dentro do conceito de sistemas eletrônicos, mas em diversas outras formas de identificação única na natureza entre outros. Por fim, estes conceitos são extrapolados para a proposta deste trabalho e desta forma, tem-se uma linha de atuação da metodologia que está apresentada nos capítulos seguintes.

No terceiro capítulo temos detalhadas as *blockchains* e o imenso universo que as moedas digitais trouxeram recentemente para inovação digital. Registra-se neste ponto, toda a origem do Bitcoin, Ethereum e outras moedas (*currencies*) digitais, ampliando o principal escopo de aplicação, que inicialmente vislumbrava a área financeira, para todo um universo de aplicações inovadoras. Explica-se neste capítulo a origem e o que é uma *blockchain*, e como ela é estruturada para implementar a rede mundial de confiança sem a necessidade de uma terceira parte para validar transações. Apresenta-se as grandes inovações que estão sendo criadas pelas diversas *blockchains* além da transformação real na vida das pessoas impactadas por esta tecnologia que muitas vezes nem sabem que ela existe. Por fim, temos a apresentação

de contratos inteligentes e de uma *blockchain* em especial que será utilizada para viabilização desse projeto, entendendo as suas vantagens, desvantagens, desafios e novas questões para a consolidação desta tecnologia.

No quarto capítulo temos apresentado o modelo que implementa o sistema proposto de autenticação de segurança em silício. Neste ponto são detalhadas todas as característica e partes funcionais da solução e justificadas cada uma delas dentro do modelo. É apresentada a solução de PUFs escolhida, suas nuances, vantagens e todos os ajustes para validação do modelo, explicando como e porque tal solução foi escolhida e como está sendo utilizada. Também são discutidos todos os parâmetros e condições de contorno com relação à *blockchain* definida e suas particularidades, apresentando as formas de implementação de contratos inteligentes e suas variantes. Por fim, temos integradas todas as soluções individualizadas em uma plataforma de autenticação e validação, que coordena todas as funcionalidades desde a operação de troca de informações dentro do componente ou circuito semicondutor, passando pela aplicação intermediária indo até o desfecho final na validação das transações de autenticação.

As conclusões e sugestões de trabalhos futuros estão apresentados também no quarto capítulo e assim conclui-se este trabalho.

2 Funções Físicas Não Clonáveis (PUFs)

No início dos anos 2000 a segurança cibernética, ou segurança digital, era apenas implementada em dispositivos eletrônicos específicos como terminais de autoatendimento, terminais de pagamentos e cartões bancários. Entretanto, atualmente qualquer transação bancária, além de diversas outras que tratam de informações sigilosas ou confidenciais, já fazem uso de recursos e de técnicas de encriptação utilizando algum tipo de algoritmo de criptografia para tentar proteger suas transações. Atualmente é comum o uso das conexões seguras de internet, "HTTPS://" *(Hyper Text Transfer Protocol Secure*), para suportarem acessos e transações com um grau maior de segurança, além do advento da assinatura digital encriptada para autenticação remota de transações, que tem contribuído muito para a segurança e validação de transações no ambiente digital.

Como consequência destas evoluções, aconteceu um crescimento muito considerável no número de circuitos integrados de aplicação específicas (ASICs), bem como, na quantidade de microcontroladores e sistemas em chips (SoCs), que embarcaram aceleradores criptográficos em "*hardware*", ou bibliotecas criptográficas de "*softwares*". Com isto e com a recente ideia da internet das coisas (*Internet of Things - IoT*) surgiu a criptografia pervasiva.

2.1 Segurança em Silício – Origem e PUFs

Entende-se como criptografia pervasiva o conceito baseado em parâmetros, regras e algoritmos conhecidos por todos, de domínio público e que rodam em diferentes plataformas, independente do hardware, software ou sistema operacional. Na criptografia pervasiva as capacidades de cifrar e decifrar a informação são permeáveis ao ambiente e podem ser realizadas por todos, tanto localmente quanto em estruturas de servidores em nuvens (*Cloud*), utilizando-se desde dispositivos móveis até de computadores de alto desempenho. Na criptografia pervasiva, a única variável que não é discreta ou disponível para o processo é a chave criptográfica. Essa chave é a figura mais importante na engrenagem da confiança da informação, afinal as demais variáveis são conhecidas e a oportunidade de garantir a segurança cibernética está baseada apenas em um componente estrutural.

Deve-se lembrar que um dos primeiros circuitos integrados (CI) projetados para proteger fortemente as chaves de criptografia teve como seu principal propósito o uso nos cartões inteligentes (*smartcards - SC*) e assim foram integrados. Com o passar dos tempos outros CIs, como os microcontroladores genéricos, foram utilizados para se implementar criptografia, com o grande desafio de garantir a proteção das chaves.

A escolha do sistema de proteção de chaves criptográficas em circuitos integrados depende de alguns fatores:

- Disponibilidade de uma tecnologia de fabricação de CI adequada: a presença de memória não volátil, como “One Time Programmable” (OTP/ROM), EEPROM ou Flash, influencia diretamente a maneira como as chaves podem ser protegidas fisicamente;
- Requisitos de mercado: o nível de segurança implementado em um CI depende de seu uso final;
- ”*Know-how*” do projetista de CI: projetar blocos de proteção de hardware ainda é uma questão de especialistas;
- Tempo de colocação no mercado;
- Custo de desenvolvimento;
- Custo por unidade (área adicional de silício).

Pode-se pensar que a proteção física não é necessária na maioria dos casos. Isso não é mais verdade, pois a engenharia reversa automatizada associada às técnicas de análise de falhas tornou os ataques físicos acessíveis [11].

A maneira tradicional de projetar o armazenamento seguro de chaves consiste em armazenar as chaves em uma memória não volátil (OTP/ROM, EEPROM ou Flash) e implementar contramedidas ou ofuscações de *layout*, como blindagem do *chip*, embaralhamento de vias ou criação de vias fictícias [12]. Uma solução mais robusta depende da criptografia de memória por meio de uma chave mestra, mas o desafio ainda é a proteção da própria chave mestra, o que retorna ao desafio inicial.

Essa abordagem é válida e é comprovada por certificações como “Critério Comum” (Common Criteria), que nada mais é do que um padrão internacional para certificação de segurança computacional. Normalmente, uma certificação “EAL4 +” [11] (nível de garantia de avaliação do “Critério Comum”), inclui o componente "AVA_VAN.5" (Análise de vulnerabilidade metódica avançada - Grau 5) [12], que classifica a resistência de um CI contra ataques físicos. Essa classificação, que é baseada na dificuldade de conduzir um ataque bem-sucedido, é apoiada principalmente nos seguintes requisitos:

- Nível de especialização;
- Tempo;
- Custo do equipamento.

Em geral, se o nível combinado dos critérios listados acima for alto o suficiente em comparação com os benefícios que o invasor obteria a implementação não será considerada válida, embora que com tempo, experiência e orçamento suficientes, ainda seja possível

recuperar as chaves.

A principal desvantagem dos métodos de ofuscação listados acima é que eles também exigem conhecimento altamente especializado, dominado por apenas alguns projetistas de CIs. Assim essas soluções não estão amplamente disponíveis e, portanto, são inaplicáveis em muitos casos.

Entretanto, como será visto mais adiante, as funções físicas não clonáveis *(Physical Unclonable Functions – PUFs)* entregues como "Propriedade Intelectual" (*Intellectual Properties – IPs*) permitem altos níveis de segurança, mesmo para não especialistas em segurança. Uma PUF nada mais é que um circuito ou função matemática que implementa uma assinatura única de um dispositivo, explorando uma característica específica que não pode ser reproduzida por nenhum outro. Uma diferença fundamental entre as técnicas tradicionais e os PUFs é que estas são, por natureza, praticamente imunes às técnicas de engenharia reversa.

Outro desafio que o PUF resolve é a necessidade de proteger as chaves antes da gravação no CI. Nas implementações tradicionais, é necessário estabelecer as chaves em alguma etapa do processo de fabricação. Isso pode acontecer no processo de fabricação do CI, diretamente no *Wafer* de silício, no teste final do CI ou na fabricação da PCB (*printed circuit board*), onde as chaves devem ser transferidas do equipamento de fabricação, ou teste para o CI. Lembrando que por se tratar de um processo de extremo risco de segurança é necessário também garantir a proteção ambiental (ambientes seguros e com controles de acesso, credenciamento e outras medidas) necessária para a integridade do processo. Gravar chaves com segurança é um processo dominado para cartões bancários, mas que pode não ser acessível para produtos médicos, industriais ou de consumo. Muitas vezes, a manufatura é realizada por subcontratados em um local remoto e é difícil exigir uma instalação segura do subcontratado, pois é necessário investir em equipamentos, redigir procedimentos e realizar auditorias periódicas.

2.2 Aplicações de PUFs

Para algorítmos criptográficos, são necessárias chaves uniformemente aleatórias e perfeitamente confiáveis. Como as respostas das PUFs geralmente são ruidosas e contém apenas uma quantidade limitada de entropia, elas não podem ser usadas diretamente como chaves. Uma etapa intermediária de processamento é necessária para obter uma chave. Esse é um problema conhecido na teoria da informação como extração de chave secreta de segredos próximos e geralmente é implementada por um algorítmo de duas fases.

PUFs em circuitos integrados têm propriedades interessantes para uso na geração e armazenamento de chaves secretas. Como a chave é gerada a partir da aleatoriedade intrínseca introduzida pela inevitável variabilidade do processo de fabricação, nenhuma etapa explícita da programação de chaves é necessária, o que simplifica a distribuição de chaves. Além disso, como essa aleatoriedade é permanentemente fixada nos detalhes físicos (*sub*) microscópicos do chip, não é necessária nenhuma memória-chave não volátil convencional. Isso também oferece segurança adicional contra ataques de sondagem e possivelmente outros ataques de canal lateral, uma vez que a chave não é permanentemente armazenada em formato digital, mas apenas aparece na memória volátil quando necessário para a operação. Além disto, possíveis evidências de violação do PUF podem ser usadas para garantir armazenamento de chaves à prova seguras, uma vez que se permite rastreabilidade da violação e a validade desta informação.

Durante a fase inicial de geração, a PUF é consultada e o algoritmo produz uma chave secreta junto com algumas informações adicionais, frequentemente chamadas de dados auxiliares. Ambos são armazenados em um local seguro pelo usuário. Na fase de reprodução, o próprio local seguro apresenta os dados auxiliares ao algoritmo que os utiliza para extrair a mesma chave do PUF que na etapa de geração. Dessa forma, o dispositivo que contém a PUF e o próprio usuário estabeleceram uma chave secreta compartilhada. É possível construir esses algoritmos de modo que a chave seja perfeitamente secreta, mesmo se os dados auxiliares forem observados, isto é, os dados auxiliares podem ser comunicados publicamente para o dispositivo.

Na Figura 2.1 têm-se os detalhes da identificação básica de uma PUF, apresentando como as funções de entrada podem gerar saídas diferentes.

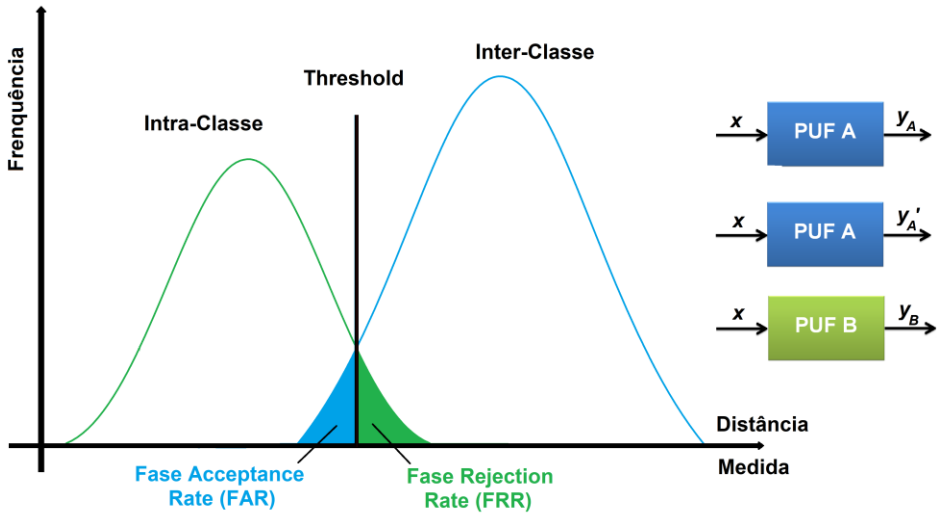

Figura 2.1 - Detalhes da identificação em uma PUF.

Como observado, é verificado que sequências de medição diferentes em instantes de tempo diferentes na mesma PUF, podem produzir assinaturas diferentes entre si, para isto é necessário entender o quão próximas duas assinaturas podem ser. Este resultado está associado com a distância intra-classe ($\mu intra$) da PUF (proximidades das respostas de saída), que analisa o quão parecido são as respostas para o mesmo desafio de uma mesma PUF (caso PUF A). A métrica usada para calcular as distâncias entre duas assinaturas é o erro absoluto normalizado entre estas assinaturas, ou seja, duas assinaturas são subtraídas uma da outra e o valor absoluto é calculado e em seguida, esse valor de diferença é normalizado pelo valor dos elementos de uma das assinaturas. Quanto menor esta diferença melhor é a PUF

No caso de PUFs diferentes no mesmo silício, a resposta dos desafios deve ser a mais diferente possível e denomina-se isto como distância inter-classe ($\mu inter$) da PUF (distanciamento das respostas de saída). Neste caso a comparação da saída entre as PUFs A e B deve ser a mais distante possível, e o processo de calculo é idêntico ao da medida da intra-classe.

Importante apontar que é possível se obter um valor de distância intra-classe maior que de inter-classe e de inter-classe menor que de intra-classe como visto na Figura 2.1, quando se trabalha com analise em baixas frequências. Estas interpolações de áreas são denominadas de taxa de rejeição de falsa (FRR) e taxa de aceitação falsa (FAR), e devem ser evitadas.

2.2.1 Proteção de IPs *Firmware/Software*

Algumas aplicações de *firmwares* e/ou *softwares* em sistemas embarcados, como por exemplo os utilizados para na área de defesa/aeroespacial, diagnóstico médico ou medição de sinais vitais, são resultados de anos de pesquisa e desenvolvimento e por consequência, de investimento pesado em recursos humanos e financeiros, encontrando-se no estado da arte da indústria de semicondutores. Assim são considerados ativos extremamente valiosos que merecem um grau de proteção muito mais forte contra engenharia reversa e/ou apropriação indevida. Neste caso, as chaves geradas por PUFs podem proteger essas IPs por meio de criptografia, usando chaves muito resilientes. Na Figura 2.2 tem-se um esquemático de como uma solução pode ser implementada, considerando além da própria PUF, também a integração de processamento (CPU) e armazenamento para o *firmware/software*. Lembrando que a PUF por si só não é capaz de gerar as chaves ou a criptografia para a proteção do dispositivo.

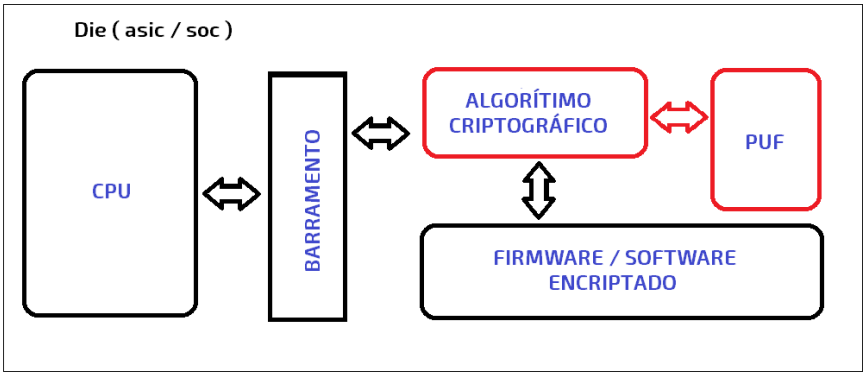

Figura 2.2 - Proteção de Propriedade Intelectual baseado em PUF

De forma simplificada podemos descrever o funcionamento desta solução da seguinte forma: primeiramente o circuito PUF gera a assinatura única extraindo-a do silício, posteriormente um Algoritmo Criptográfico, que pode ser implementado em silício (VHDL), realiza o processamento e criação da chave primária que por sua vez é usada para criptografar e descriptografar o *firmware/software* que irá ser executado no processador (CPU).

2.2.2 Armazenamento seguro de chaves privadas e secretas

Como já apresentado, o armazenamento de chaves é frequentemente a principal preocupação dos sistemas que pretendem operar com segurança e proteção de seus dados e informações. A chave gerada pela PUF neste caso é usada para criar na memória não volátil do chip, como EEPROM ou Flash, um cofre seguro habilitando assim um alto grau de confiabilidade na segurança destas informações. Atualmente esta é a técnica utilizada em carteiras de hardware de moedas digitais e em algumas soluções de *smartcards* para transações financeiras digitais. Como visto na Figura 2.3, tem-se a implementação de uma PUF que é utilizada através de um algorítmo criptográfico para habilitar ou não o acesso ao cofre seguro onde chaves e possivelmente outras informações sigilosas estão armazenadas.

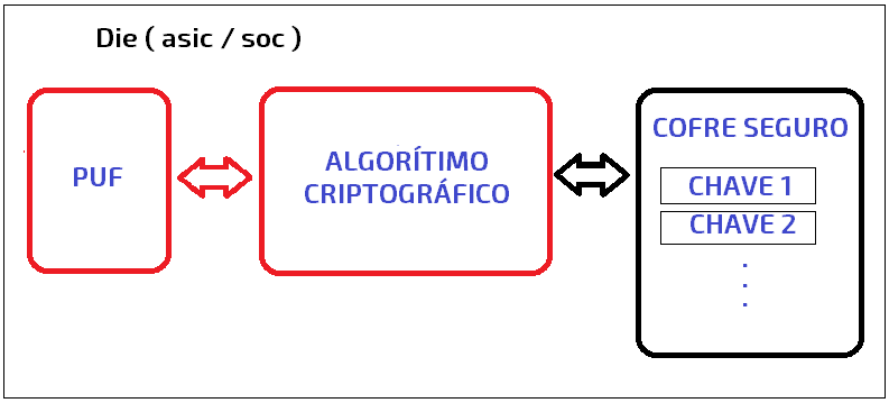

Figura 2.3 - Implementando um cofre de chaves seguras com um PUF.

2.2.3 Autenticação de dispositivos

Atualmente um dos primeiros requisitos de segurança para dispositivos conectados é a autenticação, ou seja, garantir que o componente de hardware seja original e genuíno. A forma mais segura de se realizar isto é executar a autenticação denominada “desafio – resposta”. Nesse esquema, um número aleatório ou teste é enviado ao dispositivo para ser autenticado (desafio) e o referido dispositivo assina o desafio com sua chave privada, retornando a informação válida (resposta), assim a transação autentica o componente. Neste caso, a chave privada deve ser fortemente protegida, não permitindo a sua emulação em ambientes ou sistemas que poderão tentar simula-la para falsificar uma autenticação.

Na Figura 2.4 temos a apresentação do modelo da transação “desafio resposta” que pode ser aplicado em uma PUF para validar a sua originalidade. Neste caso em particular o desafio e a resposta podem ser implementados e obtidos de forma interna ou externa ao chip, utilizando, por exemplo, plataformas de sistemas automáticos para acesso, verificação e autenticidade do componente semicondutor.

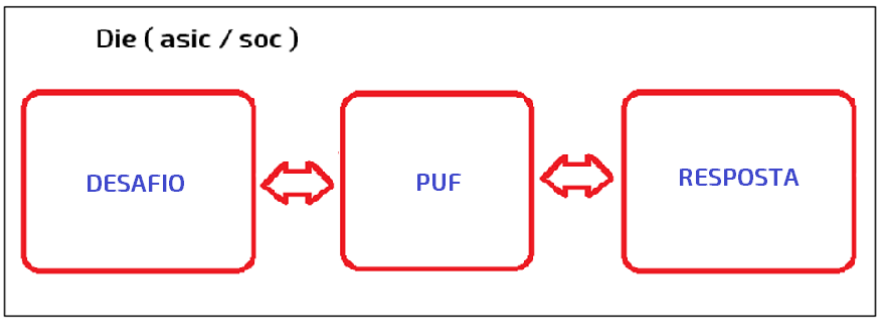

Figura 2.4 - Modelo “Desafio-Resposta” com um PUF.

2.3 Caracterizando PUFs

Segundo [13], algumas PUFs estão sendo contestadas na literatura com relação a certas propriedades de segurança devido a possibilidade com novos recursos tecnológicos de replicação da sua assinatura, porém deve-se considerar as principais propriedades necessárias para o uso de PUFs em aplicações de segurança e criptografia, como por exemplo:

- Não clonabilidade: Nenhum processo físico eficiente é conhecido que permite a clonagem física de PUFs, ou seja, cada PUF é única;
- Evidência de violação: Violações físicas na PUF provavelmente irão destruir a sua estrutura física, tornando-a inútil, ou transformando-a em uma nova PUF;
- Independência: Duas diferentes PUFs apresentam comportamentos

completamente independentes;

- Distribuição não uniforme: As saídas da PUFs não são uniformes, ou seja, a distribuição de probabilidade não é igual.

Deve-se considerar também que uma função física não clonável pode ser obtida não apenas pela lógica combinacional de circuitos, mas por vários mecanismos que envolvem variações de processo, ou seja, qualquer elemento natural ou artificial que pode ser criado por um processo de repetição, pode produzir uma PUF, uma vez que praticamente quase tudo na natureza possuirá um variação, mesmo que mínima, entre iguais.

Assim, serão apresentados alguns tipos de PUFs que ilustram intuitivamente como o processo de variabilidade de propriedades físicas podem produzir componentes diferentes.

2.3.1 PUFs não eletrônicas

PUFs óticas: estas funções físicas não clonáveis podem ser observadas nas características de transparência de um determinado alvo e foram propostas em [14] [15]. Basicamente, como visto na Figura 2.5, existe um processo desafio-resposta onde um sistema laser projeta um feixe em um alvo transparente (*token*) e assim, devido às configurações do laser e a mecanismos de posicionamento do sistema, existe uma leitura de retorno, não dependente de injeção de sinal elétrico. O *token* ótico contém uma microestrutura de refração em uma pequena placa transparente, que ao ser atingido responde com uma assinatura única, onde através de um dispositivo como um sensor "*charge coupled device* - CCD" (câmera) permite extrair um padrão de mancha. Através deste processo, com o uso de um gerador de código *Hash*, se produz um padrão de resposta que comparado com o desafio proposto pelo *token* (processamento da informação), autentica o dispositivo.

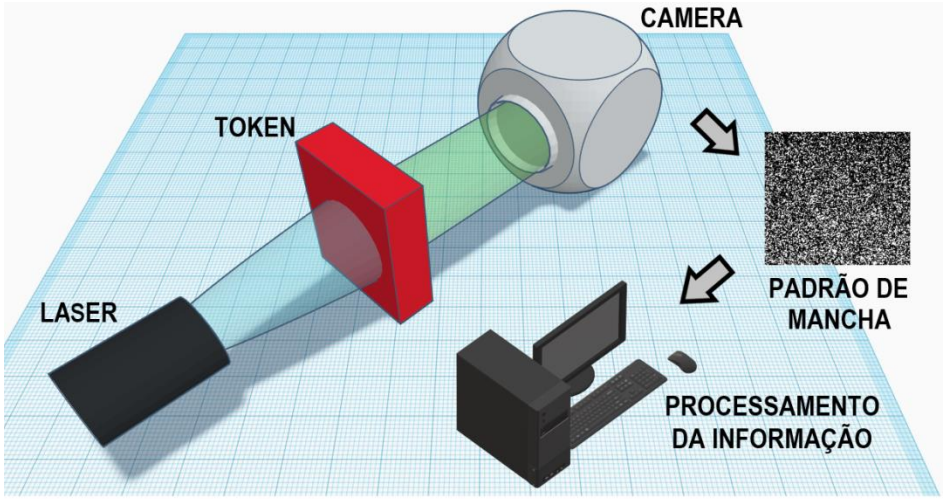

Figura 2.5 - Operação genérica de um PUF óptico.

Como curiosidade vale lembrar que as câmeras digitais possuem sensor CCD e também podem ser identificadas devido a variações mínimas de processo de fabricação deste componente semicondutor. Essa característica pode, por exemplo, identificar qual câmera digital foi usada para tirar uma determinada foto e assim relacionar o hardware à imagem produzida pelo dispositivo.

PUFs de papel: A PUF de papel é na verdade uma proposta feita na literatura, que consiste no escaneamento da estrutura de fibras únicas e aleatórias de um papel regular ou modificado, apresentando um padrão não clonável. No caso das PUFs de papel em [16] [17], existem várias propostas de uso antes mesmo do conceito moderno de PUF, que são hoje aplicadas principalmente como uma estratégia antifalsificação de papel moeda. Seu processo de obtenção é realizado pela reflexão de um feixe de laser focalizado em uma parte do papel utilizado em um documento, extraindo-se assim, a "impressão digital" desse documento pelo comportamento e direções de suas fibras. Uma abordagem semelhante é usada em [18], mas se introduzem neste caso, fibras ultravioletas no papel durante o processo de fabricação que posteriormente podem ser medidas por um *scanner* convencional. Nesta proposta, também se introduz um método para vincular os dados do documento ao papel em específico, usando uma combinação de assinatura digital dos dados e a impressão digital do papel impresso pelo documento.

Outras PUFs que podem ser citadas nesta categoria são: PUFs de *Compact Disc* ou CD onde se observa que os comprimentos medidos de pistas (trechos de poço e pico) em um CD, contêm um desvio aleatório em relação aos comprimentos pretendidos e podem ser utilizados para uma identificação única deste dispositivo. PUFs acústicas: nesta proposta, linhas de atraso acústicas podem ser usadas para atrasar sinais elétricos. Eles convertem um sinal elétrico alternado em vibração mecânica e vice-versa, e são construídas observando o espectro de frequência característico de uma linha de atraso acústica [19].

2.3.2 PUFs de eletrônica analógica

PUFs de VT (tensão de *threshold*): esta provavelmente foi a primeira técnica para se tentar produzir uma identificação única em um circuito integrado convencional, sem a necessidade de etapas especiais de processo de fabricação, ou programação de componentes pós-fabricação. Na ICID [20], o princípio básico de operação é relativamente simples. Um número de transistores igualmente projetados são dispostos em uma matriz endereçável. O

transistor endereçado aciona uma carga resistiva e, devido ao efeito das variações de fabricação nas tensões limiares (VT) desses transistores, a corrente através dessa carga será única. A tensão sobre a carga é medida e convertida em uma sequência de bits com um comparador de zeramento automático, produzindo assim um processo de assinatura randomizada. A técnica foi verificada experimentalmente em 55 chips produzidos em tecnologia CMOS 0,35 um [20].

PUFs de revestimento: Na proposta [21] se abrigava sensores em forma de pente na camada superior de metal de um circuito integrado e assim se iniciava a construção de um sistema integrado entre estes dispositivos. Essa proposta considera a aleatoriedade das medições de capacitância nestes sensores. Apresentada na Figura 6, além dos efeitos aleatórios da variabilidade produzida pelo processo fabricação, outros elementos aleatórios foram incluídos por meio de um revestimento dielétrico passivo pulverizado diretamente sobre os sensores, desta forma, mais um grau de randomização elétrica foi adicionado. Como o revestimento é opaco e quimicamente inerte, oferece forte proteção contra ataques físicos, além de proteção mecânica ao circuito integrado. Uma avaliação experimental de segurança revelou que a PUF de revestimento também é considerada inviolável no processo de engenharia reversa ou observação invasiva, ou seja, após um ataque qualquer com um feixe de íon focalizado (focused ion beam – FIB), as respostas do PUF são significativamente alteradas, modificando assim a assinatura original. Nesta proposta foram produzidos 36 chips, cada um com 31 sensores, e os resultados das medições mostraram alta aleatoriedade e baixo ruído, após quantização.

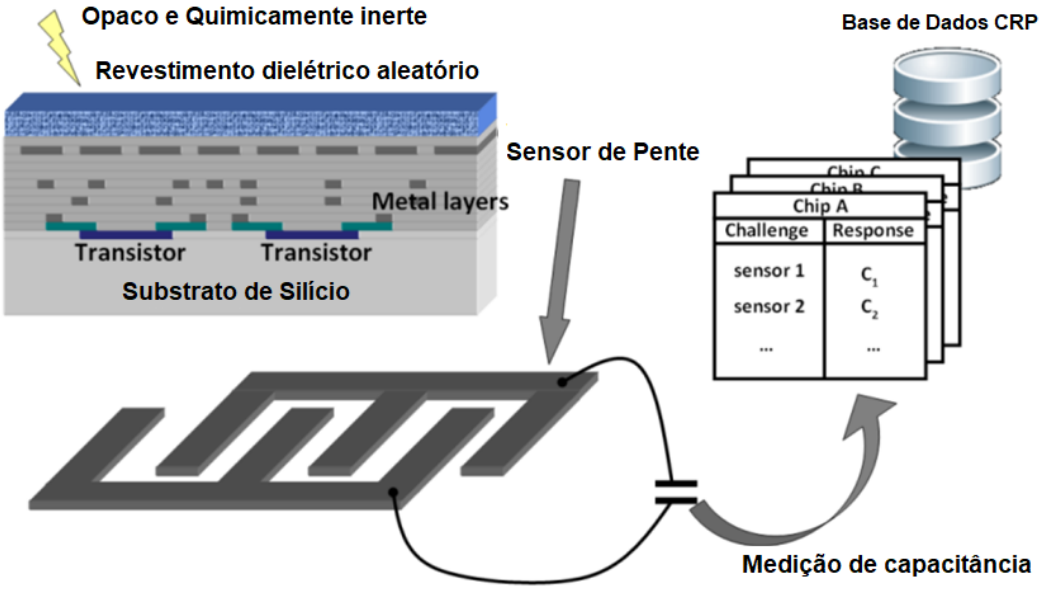

Figura 2.6 - Operação genérica de um PUF de revestimento.

2.3.3 PUFs baseadas em atraso intrínseco (*delay-based intrinsic*)

Nas PUFs apresentadas, inicia-se com uma medida analógica de um parâmetro físico aleatório, que é posteriormente quantizado e pode ser usado como um identificador de todo o sistema. Embora a definição não formal de uma PUF baseadas em atraso intrínseco seja fornecida na literatura, se faz necessário a distinção de dois pré-requisitos para que a PUF seja chamada de intrínseca.

- PUF, incluindo o bloco de medição, deve estar totalmente integrado no dispositivo embarcado;
- A construção completa da PUF deve consistir de primitivas que estão naturalmente disponíveis para o processo de fabricação do dispositivo embarcado.

Sendo assim, a primeira condição implica que o dispositivo possa consultar e ler seu próprio PUF sem a necessidade de instrumentos externos e sem a necessidade de que o conjunto "desafio/resposta" para sua leitura precise ser feita fora do dispositivo. Alguns exemplos anteriormente apresentados já atendem a essa condição, como a PUF de revestimento ou a versão integrada da PUF óptica. A segunda condição implica que a construção completa da PUF não possui praticamente nenhuma sobrecarga adicional além do espaço ocupado pela PUF, ou seja, nenhuma etapa extra de fabricação ou componentes especializados se fazem necessários. Isso não vale mais para a PUF de revestimento e a PUF com óptica integrado, pois ambas precisam de etapas de processamento altamente especializadas fora da estrutura intrínseca.

A grande vantagem de uma PUF integrada em um chip digital é que as respostas da PUF podem ser usadas diretamente por outros aplicativos em execução no mesmo dispositivo. Várias PUFs intrínsecas já foram propostas e todas integradas em circuitos digitais. Entretanto, ha duas classes diferentes de PUFs intrínsecas; PUFs intrínsecas com base em medições de atraso digital e PUFs intrínsecas com base na determinação de elementos de memória (esta última será abordada na seção seguinte). A seguir dois modelos baseados em atraso.

PUFs baseada em árbitro: A proposta inicial de uma PUF baseada em árbitro foi feita em [22] [23]. A ideia básica é introduzir uma condição de corrida digital entre dois caminhos em um chip e fazer com que o chamado circuito árbitro decida qual dos dois caminhos venceu a corrida. Se os dois caminhos são projetados simetricamente, ou seja, com o mesmo atraso pretendido, o resultado da corrida não é previamente determinado. Durante a produção do chip, as variações de fabricação afetam os parâmetros físicos, determinando o atraso exato de

cada caminho e causando um pequeno conjunto aleatório entre os dois atrasos. Isso leva a um resultado aleatório e possivelmente específico do dispositivo do árbitro e, portanto, explica o comportamento da PUF dessa construção. Se o conjunto de chaves de percurso for muito pequeno, o tempo de espera do circuito do árbitro será violado e sua saída não dependerá mais do resultado da corrida, mas será determinada por ruído aleatório. Este último fenômeno é chamado de metaestabilidade do árbitro e introduz ruído nas respostas do PUF.

Na Figura 2.7, o projeto proposto usa os chamados blocos de comutação para construir dois caminhos simétricos e um flip-flop para implementar o circuito do árbitro.

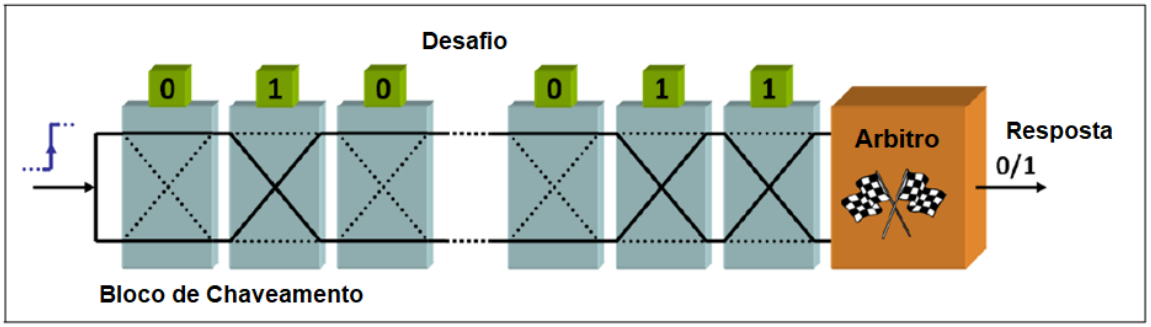

Figura 2.7 - Operação de um PUF de árbitro.

Cada um dos blocos de chave possui duas entradas e duas saídas e, com base no bit de parâmetro, eles são conectados diretamente ou comutados. A conexão de vários blocos de comutação em série cria duas linhas de atraso parametrizáveis que alimentam o árbitro. A configuração dos blocos de chave será o desafio do PUF, e a saída do árbitro a resposta. Observe que o número de possíveis desafios é exponencial em função dos números de blocos de comutação utilizados.

Esse projeto [22] [23] foi implementado em um ASIC, encadeando 64 blocos de comutação. A validação experimental em 37 chips mostra $\mu inter$=23% e $\mu intra$ < 5%, mesmo sobre variações consideráveis de temperatura e tensão de alimentação. Estes mesmos testes implementados em FPGA mostraram muito menos aleatoriedade única, ou seja, desvio de inter-classe e intra-classe ($\mu inter$=1,05%; $\mu intra$=0,3%), provavelmente devido às restrições de roteamento discretas, implícitas na arquitetura da FPGA.

Simultaneamente com a introdução de PUFs baseadas em atraso, foi reconhecido que o atraso digital é aditivo por natureza. No caso do árbitro PUF de [22] [23], o atraso da cadeia de blocos de comutação será a soma dos atrasos dos blocos separados. Essa observação leva aos chamados ataques de construção de modelos, ou seja, é possível construir um modelo matemático do PUF que, após observar várias consultas de atraso, é capaz de prever a resposta a um desafio invisível com precisão relativamente alta. Todo o trabalho subsequente

em PUFs de árbitros é basicamente uma tentativa de reduzir a efetividade dos ataques de construção com modelos mais difíceis, introduzindo não linearidades nos atrasos, controlando e/ou restringindo as entradas e saídas do PUF.

PUFs baseadas em osciladores de anel: As PUFs baseadas em oscilador de anel usam uma abordagem diferente para medir pequenos desvios aleatórios de atraso causados pela variabilidade da fabricação [24] [25]. A saída de uma linha de atraso digital é invertida e retornada à sua entrada, criando um loop de oscilação assíncrona, também chamado de oscilador de anel. É evidente que a frequência deste oscilador é determinada com precisão pelo atraso exato da linha de atraso. Medir a frequência é, portanto, equivalente a medir o atraso, e devido a variações aleatórias de fabricação, a frequência exata também será parcialmente aleatória e dependente do dispositivo. As medições de frequência podem ser feitas com relativa facilidade usando componentes digitais: um detector de rampas (*edge detector*) detecta transições positivas na oscilação periódica e um contador digital conta o número de transições durante um período de tempo.

O valor do contador contém todos os detalhes da medida desejada e é considerada a resposta PUF. Se a linha de atraso for parametrizável como no design básico da PUF baseada em árbitro, a configuração de atraso específico será novamente considerado o desafio. Os blocos básicos de construção do oscilador de anel simples são o detector de rampas e o contador digital.

A Figura 2.8 exemplifica a construção de uma PUF baseada em osciladores de anel. Nela pode-se observar a realimentação do sistema, o detector de rampas e o contador digital, além de todo o fluxo de transmissão do sinal produzido no circuito de atraso e posteriormente detectado e registrado no contador.

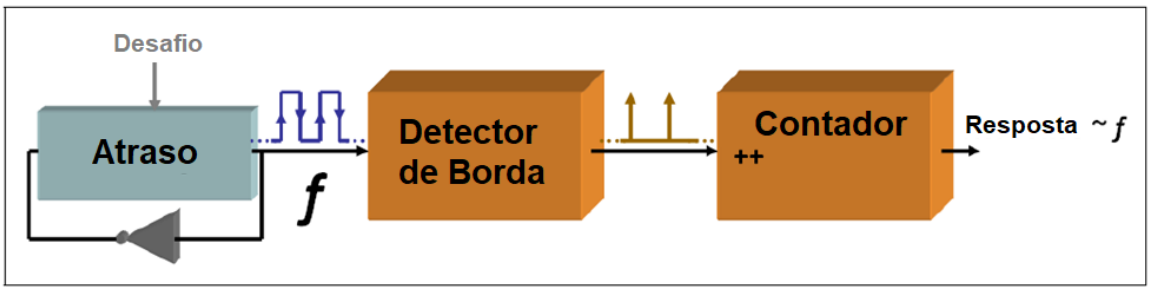

Figura 2.8 - Operação básica de um PUF de oscilador em anel.

Conforme explicado anteriormente, alguns parâmetros ambientais podem afetar indesejadamente as respostas da PUF. No caso de medições de atraso em circuitos integrados, a temperatura e a tensão de alimentação afetam fortemente o atraso exato. Para PUFs de árbitros, esse efeito não é tão grande, pois implicitamente executam uma medição diferencial

considerando dois caminhos de atraso paralelo simultaneamente.

Para PUFs de oscilador de anel, esses efeitos são muito maiores e é necessário algum tipo de compensação [24] [25]. A técnica consiste em dividir os valores de contador de duas medidas simultâneas, o que leva a respostas muito mais robustas. Nesta proposta [24] [25] foi testada uma PUF de oscilador de anel com compensação de divisão em 4, obtendo $\mu inter$=10x10^{-3} e $\mu intra$=0,1x10^{-3} com medições realizadas em uma temperatura de 25C. Também foi mostrado que as variações de tensão de alimentação aumentam $\mu intra$ com uma variação de 0,003x10^{-3} por variação de mV.

2.3.4 PUFs baseadas em estados intrínsecos de memórias

Uma célula de memória digital é tipicamente um circuito digital com mais de um estado logicamente estável. Ao residir em um de seus estados estáveis, pode armazenar informações, por exemplo, um dígito binário no caso de dois possíveis estados estáveis. Pensando no dado armazenado, se o elemento entrar em um estado instável, não está claro o que acontecerá. Pode começar a oscilar entre estados instáveis ou pode convergir de volta para um de seus estados estáveis.

No último caso, observa-se que determinadas células preferem fortemente certos estados estáveis em detrimento de outros. Além disso, esse efeito geralmente não pode ser explicado pela implementação lógica da célula, mas acontece que a instabilidade pode ser causada pela variação de fabricação. Por esse motivo, o grau de estabilização de uma célula de memória desestabilizada é um bom candidato para uma resposta de uma PUF. Há diferentes propostas da literatura, com base em diferentes tipos de células de memória, como células SRAM, *latches* e *flip-flops*.

SRAM PUFs: As PUFs SRAM propostas em [26] são um conceito muito semelhante ao que foi apresentado em [27]. SRAM ou memória estática de acesso aleatório é um tipo de memória digital que consiste em células capazes de armazenar dígitos binários. Uma célula SRAM, como mostrado na Figura 2.9, é logicamente construída como dois inversores de acoplamento cruzado levando, portanto a dois estados estáveis. Na tecnologia CMOS comum, esse circuito é implementado com 4 MOSFETs e outros 2 MOSFETs são usados para acesso de leitura/gravação, como mostra a Figura. 2.10. Por razões de desempenho, a incompatibilidade física entre as duas metades simétricas do circuito (cada um dos inversores) é mantida a menor possível. Não se sabe a partir da descrição lógica da célula em que estado ela estará na inicialização, ou seja, o que acontece quando é energizada.

Observa-se que algumas células preferencialmente armazenam zero, outras preferencialmente armazenam um e algumas células não têm preferência real, mas a distribuição desses três tipos de células na memória é completamente aleatória.

Assim, a incompatibilidade física aleatória na célula, causada pela variabilidade de fabricação, determina o comportamento de inicialização, forçando uma célula a zero ou um durante a inicialização, dependendo do sinal da incompatibilidade. Se a incompatibilidade for muito pequena, o estado de inicialização é determinado pelo ruído estocástico no circuito e será aleatório sem uma preferência real.

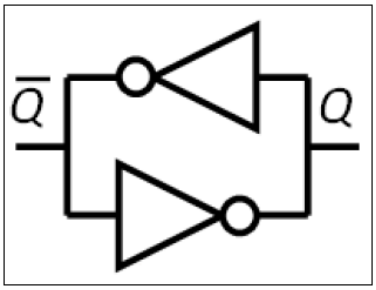

Figura 2.9 - Circuito Lógico de uma célula SRAM (PUF).

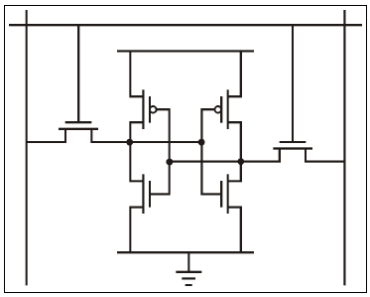

Figura 2.10 - Circuito Elétrico de uma célula SRAM (PUF) em tecnologia padrão CMOS.

Em [26], foram realizadas extensas experiências com PUFs SRAM. Foram coletados os estados de inicialização de 8190 bytes de SRAM de diferentes blocos de memória em diferentes FPGAs. Os resultados mostram uma inter-classe média entre dois blocos diferentes de $\mu inter$= 49,97% e a intra-classe média em várias medições de um único bloco de $\mu intra$= 3,57% para um ambiente fixo e $\mu intra$<12 % para grandes desvios de temperatura.

PUFs *Latch*: PUF *latch* é uma técnica de identificação de CI proposta em [28] que é muito semelhante às PUFs SRAM e *Butterfly* PUFs [29]. Em vez de acoplar dois inversores ou dois *latches*, duas portas NOR são acopladas como mostrado na Figura 2.11, constituindo um *latch* NOR simples. Ao afirmar um sinal de redefinição (inicialização), esse *latch* se torna instável e converge novamente para um estado estável, dependendo da incompatibilidade interna entre os componentes eletrônicos. Uma célula *Butterfly* PUF é um circuito biestável de acoplamento cruzado, que pode ser levado a um estado instável antes de se estabelecer em um

dos dois estados estáveis possíveis. A estrutura é composta por dois *latches* cujas saídas são cruzadas conforme indicado na Figura 2.12.

Experimentos em 128 NOR *latches* implementados em 19 ASICs fabricados com tecnologia CMOS de 0,130 µm resultaram em $\mu inter$= 50,55% e $\mu inrar$= 3,04%.

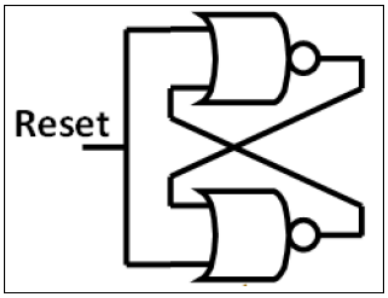

Figura 2.11 - Circuito Lógico de um *latch* (PUF).

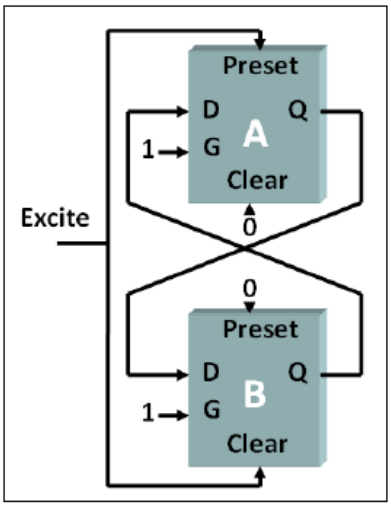

Figura 2.12 - Circuito Esquemático de uma célula *Butterfly* (PUF).

PUFs *Flip-flop*: observando PUFs SRAM, o comportamento das PUFs *flip-flop* pode ser extrapolado, pois o funcionamento é muito parecido. Nestes arranjos se exploram os valores de inicialização de um *flip-flop* D configurável. Experimentos conduzidos por [29] resultaram para 4096 *flip-flops* de 3 FPGAs, $\mu inter$= 11% e $\mu intrar$= <1%.

2.3.5 Melhorar uma PUF (usando técnica de controle - *hash*)

Como dito, uma PUF extrai informações de identificação de um CI e em princípio acredita-se que essa informação seria difícil de ser prevista por um invasor. Entretanto em propostas mais robustas, podem se fazem necessárias em outras implementações. Ao adicionar uma técnica de controle a uma PUF, é possível torná-la muito mais forte.

Neste exemplo, apesar de não utiliza-la efetivamente na demonstração e validação desta tese, é importante conhece-la. Tem-se uma PUF *f* que se deseja melhorar de alguma forma. Técnicas de controle permitem melhorar f construindo uma nova PUF *g*, baseado em *f*. Nesta solução, o controle só permite que *f* seja observado como parte de uma avaliação de *g* (saída geral do sistema), e só usa o resultado da observação para ajudar a avaliar *g*.

O diagrama de blocos na Figura 4.13 mostra essa proposta de melhoria. Para que as melhorias sejam robustas ao ataque físico, a lógica que cerca a PUF *f* original deve estar entrelaçada com a nova PUF *g*, para que um invasor não possa contornar a lógica por meio de uma sondagem física. Em particular, ele deve ser impedido de ler a resposta da PUF *f* original diretamente, antes de passar pela função de controle aleatória (*hash*) de saída e de contornar a função aleatória (*hash*) de entrada conduzindo o desafio diretamente a PUF *f*.

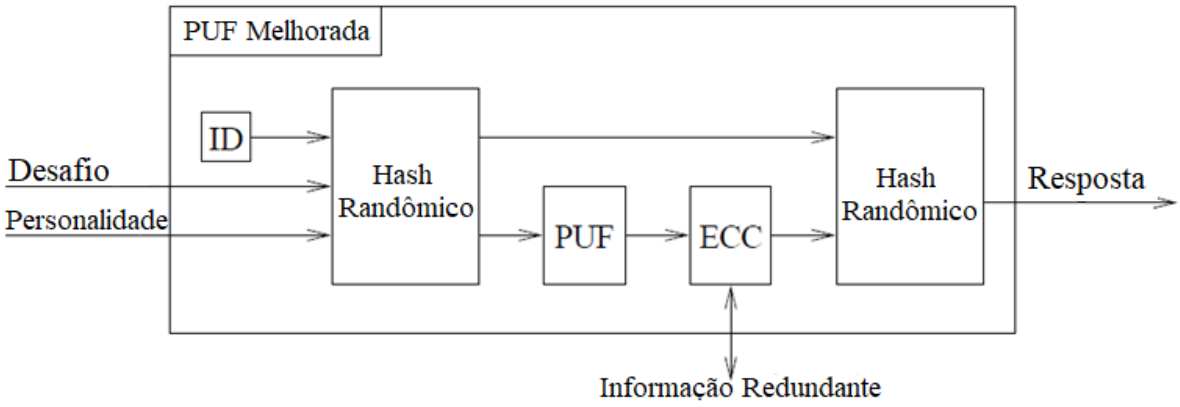

Figura 2.13 - Melhorias para robustez de uma PUF *f* (intern) e uma PUF *g* (Improved).

O número de parâmetros físicos que definem uma PUF é proporcional ao tamanho do sistema que a define. Portanto, em princípio, se um invasor é capaz de determinar uma série de parâmetros primitivos que são proporcionais ao tamanho do sistema físico, ele pode usá-los para simular o sistema e assim clonar a assinatura da PUF.

Para tentar determinar os parâmetros primitivos, o invasor precisará obter uma série de pares de desafio-resposta (*Challenge Response Pairs* - CRPs) e os usar para construir um sistema de equações que lhe permita tentar resolver o CRP. Por definição, para uma PUF, essas equações são impossíveis de se resolver em um tempo razoável. No entanto, apesar de haver sistemas físicos para os quais a maioria dos CRPs leva a equações insolúveis, existe um pequeno subconjunto de CRPs que pode fornecer equações que são capazes de quebrar a PUF, o que consequentemente, não a qualificaria realmente como uma verdadeira PUF.

Sendo assim, tal sistema não seria seguro porque um invasor poderia usar os CRPs que levam a equações simples para obter um conjunto de equações solucionável e então, calcular

os parâmetros primitivos e clonar o PUF construindo assim um simulador. Com controle por *hash*, é possível construir um sistema seguro a partir de uma dessas PUFs "quebráveis". Uma maneira de se conseguir isro é fazer com que a camada de controle simplesmente se recuse a dar respostas aos desafios que levam a equações simples. Infelizmente, esse método pressupõe que se conheçam todas as estratégias que o invasor pode usar para obter um conjunto simples de equações de um conjunto escolhido de CRPs.

Extração da Codificação Randômica - É desejável que a saída de uma PUF exiba o máximo de aleatoriedade possível para evitar que um invasor adivinhe a resposta a um desafio usando a resposta a outro desafio. No entanto, a saída de um sistema físico provavelmente produzirá respostas semelhantes quando confrontado com estímulos semelhantes. Além disso, os CRPs podem ser usados para se obter sistemas de equações que relacionam os parâmetros físicos subjacentes da PUF.

Ambos os riscos podem ser eliminados fazendo uma transformação simples, se f é a PUF original que se tenta melhorar e h é uma função *hash* aleatória, então a nova PUF g a seguir **"$\mathbf{g_{(x)} = h_{(x,f(x))}}$"** é uma PUF mais forte. O efeito de avalanche da função *hash* garante que as saídas próximas de f produzam saídas completamente diferentes da função composta, e a natureza unilateral de h significa que para configurar um sistema de equações, o invasor terá que inverter h (ou incluir a definição de h no sistema de equações, que é tão improvável quanto a primeira).

A pós-composição da PUF com uma função *hash* é uma etapa muito importante porque torna o sistema comprovadamente resistente a ataques não físicos, desde que informações suficientes sejam extraídas do sistema físico antes de executá-lo por meio da função aleatória de saída. No caso de um circuito de atraso, a abordagem certa seria medir um número de atrasos até que algumas centenas de bits tenham sido extraídos do sistema e então, executar todos eles por meio da função de *hash*.

Dando a uma PUF Múltiplas Personalidades - Na verdade, a experiência anterior mostra que os usuários se sentem desconfortáveis com processadores que possuem identificadores exclusivos, porque sentem que podem ser rastreados. Os usuários podem ter o mesmo tipo de preocupação com o uso de PUFs, visto que as PUFs são uma forma de identificador único.

Este problema pode ser resolvido fornecendo uma PUF com múltiplas personalidades. O proprietário da PUF possui um parâmetro que pode controlar, a que lhe permite mostrar diferentes facetas de seu PUF para diferentes aplicações. Para fazer isso, implementa-se uma função *hash* desafio como um número de personalidade selecionado pelo usuário e usa-se essa

hash como entrada para o resto da PUF.

Desse modo, o proprietário efetivamente tem varias PUFs diferentes à sua disposição, de modo que a terceiros aos quais ela mostrou funções aleatórias (*hash*) terão apenas funções de caminho único, e com diferentes personalidades não podendo determinar se interagiram com a mesma PUF original ou com uma função de embaralhamento.

Correção de Erro - Em muitos casos, a PUF está sendo calculada usando um sistema físico analógico. É inevitável que pequenas variações de uma execução para a outra podem causar pequenas alterações na saída digitalizada da PUF. Isso significa que o chip produz uma aproximação da resposta que se espera dele. Em algumas aplicações, o chip e o desafiante não podem comparar diretamente a resposta real com a resposta desejada, pois isto exigiria o envio de uma das respostas em aberto, comprometendo assim o segredo compartilhado. Portanto, algo deve ser feito para tornar a saída da PUF idêntica cada vez que um desafio for reutilizado.

Um código de correção de erros adequadamente selecionado é uma possibilidade. Quando um par desafio-resposta é criado, algumas informações redundantes também são produzidas, o que deve permitir que pequenas variações nos parâmetros medidos sejam corrigidos. Nos usos subsequentes do par desafio-resposta, a informação redundante é fornecida ao PUF junto com o desafio. Isto é usado para se corrigir a resposta do sistema físico.

Naturalmente, a correção de erros deve ocorrer diretamente nos parâmetros físicos medidos. Em particular, se a PUF for pós-composta com uma função aleatória (*hash*), a correção deve ocorrer primeiro. Se várias medições estiverem sendo combinadas em uma resposta, a correção de erros deve operar em todas as medições.

Rodadas múltiplas - Para adicionar ainda mais complexidade ao problema de um possível invasor, seria possível usar o circuito PUF várias vezes para produzir uma resposta. A resposta corrigida de uma rodada pode ser realimentada no circuito PUF. Depois de algumas rodadas executadas, todas as suas saídas poderiam ser mescladas junto com o desafio, a personalidade e o identificador do chip por uma função *hash* aleatória para produzir a resposta global.

Identificador Único - No caso das PUFs originárias por fabricação, tem-se esta funcionalidade obtida pela limitação do controle do fabricante sobre as variações do processo de fabricação. Cada PUF é diferente devido a essas variações, no entanto, é possível que existam PUFs idênticas. Isso não é um grande problema, porque, em geral, encontrar um par de PUFs idênticas requer a produção e a comparação de um número muito elevado de PUFs.

No entanto, é possível garantir que duas PUFs sejam sempre diferentes e para fazer isso, combina-se o desafio real e um identificador exclusivo do chip, com uma função *hash* antes de executá-los no restante da PUF. O identificador único não precisa ser secreto e pode ser, por exemplo, o número de série do circuito integrado.

Desta forma, não há duas PUFs iguais, e mesmo se duas PUFs compartilhassem a mesma PUF f subjacente, não há como um invasor descobrir isso. O fabricante pode ser capaz de descobri-lo antes de definir o identificador exclusivo da PUF, mas o custo do teste é proibitivo em qualquer caso, pois seriam testadas, como dito antes, um número muito elevado de chips, para tentar encontrar PUFs idênticas.

2.4 Proposta

Este capítulo apresentou as necessidades de proteção e segurança da informação nos tempos atuais. Com isto, foram relacionadas algumas demandas e a principal proposta de trabalho, que é o uso de funções físicas não clonáveis – PUFs.

Foi necessário abordar todos os quesitos que tornam esta tecnologia fundamental para se garantir um alto nível de proteção aos dispositivos que fazem uso desta solução. Além disto, registrou-se um apanhado sobre PUFs em suas mais diversas formas, com a intenção de nivelamento de conhecimento por parte dos que farão uso deste trabalho. Somente assim será possível entender do porque, usar esta solução integrada com outras tecnologias, pode trazer resultados efetivos para o tema de segurança cibernética.

Vale aqui ressaltar que o uso de PUFs não é uma solução nova e já vem sendo utilizada a algum tempo em soluções para sistemas e componentes seguros, porém será apresentada uma solução, integrando estes recursos com outras soluções no itém 4 referente ao modelo objeto deste trabalho.

Por fim, a intenção aqui é resgatar o conhecimento, apresentar as diversas variantes das funções físicas não clonáveis, e como estas podem ser utilizadas na segurança e autenticação de dispositivos. Lembrando que dentre as soluções apresentadas, serão tratadas principalmente das PUFs de atraso, onde se detalhará uma implementação para o projeto que trata do modelo proposto.

3 *BLOCKCHAIN* – O protocolo da confiança

Blockchain é uma base de dados descentralizada, controlada e verificada por todos os atores que necessitam transacionar ativos digitais na internet ou implementar soluções de registro e validação distribuída. A *blockchain* permite que uma entidade transacione diretamente com outra, sem a necessidade de uma autoridade centralizada autenticadora de transações.

3.1 Origem

O Bitcoin é uma moeda digital que foi criada e estruturada sobre uma *blockchain* para permitir atribuir valor a transações em um ambiente digital. O Bitcoin não é o único ativo digital existente hoje na *internet* que faz uso de uma *blockchain*, mais com certeza é o mais valioso e também mais famoso.

A *blockchain* é a essência do protocolo Bitcoin, proposto por Satoshi Nakamoto [31], que entrou em operação em 2009. O artigo incialmente proposto descreve uma rede ponto a ponto (P2P) onde transações com a criptomoeda Bitcoin, são recebidas por servidores descentralizados (mineradores), que através de processamento em tempo real, irão validar esta operação. O processo é realizado através de um protocolo específico de consenso à base de desafios criptográficos (checagem de função de *hash*), que testam a validade de cada transação e a ordem em que as mesmas serão permanentemente armazenadas em uma corrente de blocos (*blockchain*) replicada na internet em cada nó ou servidor que colaborou para processa-la após o consenso de validação de todos.

O ponto focal da crescente expectativa pelo uso do Bitcoin é a chamada “Prova de Conceito” (POC) e foi fundamental para uma crescente adesão à rede de confiança digital descentralizada que se criou a partir desta ideia. O grande diferencial: eliminar a terceira parte de confiança, necessária nas transações financeiras e assim liberar o mundo digital para transacionar ativos diretamente entre contas. Para além da tecnologia, a *blockchain* do Bitcoin traz uma ruptura nas transações de negócios, ao introduzir o mecanismo de incentivos digitais e criptomoedas em diversos níveis de relações. De fato o consenso de Nakamoto (com sua mineração inovadora) e a geração da moeda (como mecanismo de incentivo) integram-se numa simbiose salutar para viabilizar o acordo entre pares que não se conhecem, e a oferta de serviços de consenso sob demanda na *Internet*.

No caso específico da *blockchain* original do Bitcoin, esta incorpora uma máquina de estados bem simplificada, com elementos voltados essencialmente para transações com a moeda Bitcoin. Por outro lado a moeda digital Ethereum, idealizado por Vitalik Buterin [32]

adota o conceito de contratos inteligentes, que permite a execução de uma máquina de Turing completa. Os contratos inteligentes expressam uma lógica de transações mais sofisticada, possibilitando a implementação de aplicações descentralizadas e autônomas, em diversos níveis e escopos.

Na evolução de projeto da blockchain, são destacadas três fases [33]: a *blockchain* 1.0 corresponde ao lançamento do Bitcoin em 2008, com as primeiras implementações das criptomoedas, e um ecossistema de aplicações e pagamentos com o ativo digital. A *blockchain* 2.0 iniciou-se com a proposta inovadora dos contratos inteligentes em 2013, e toda gama de aplicações financeiras possíveis. Já *blockchain* 3.0 é caracterizada pela adoção da tecnologia *blockchain* no benefício de aplicações em diversas áreas, para além da financeira: governo, comércio, artes, saúde, cidades digitais, etc. É nesta fase 3.0 que este trabalho se desenvolve e propõe inovações na área de autenticação e segurança cibernética.

Há diversos projetos, com forte investimento da indústria, visando a implementação de plataformas de *blockchain* para desenvolvimento de aplicações robustas e descentralizadas nos mais variados segmentos. Para além do Bitcoin e Ethereum, que oferecem uma *blockchain* pública, aberta, sem necessidade de controle dos participantes, há uma gama de setores que requerem participação controlada. As plataformas para *blockchain* privada ou federada atendem melhor a interesses corporativos e requerem a autenticação dos participantes na rede; dentre elas, destaca-se o Hyperledger Fabric [33], em estágio avançado de desenvolvimento.

3.2 Histórico

Vale registrar aqui um pouco da história deste processo apresentando cronologicamente o Bitcoin e a tecnologia da *blockchain* [34].

TIME LINE	BITCOIN	BLOCKCHAIN
2009 2014	PRIMEIRA TRANSAÇÃO DE BITCOIN - Satoshi Nakamoto registra a primeira transação de Bitcoin. MERCADO DE BITCOIN - O Bitcoin Market se torna a primeira bolsa oficial de criptomoedas. A MINERAÇÃO DE BITCOIN - Uma Industria em crescimento - Artforz cria a primeira 'fazenda' de mineradores de Bitcoin. BTC / $ PARITY - O valor do Bitcoin atinge a paridade com o dólar americano	Blockchain está para Bitcoin, o que a Internet está para e-mail. Um grande sistema eletrônico sobre o qual pode-se construir aplicativos. A moeda é apenas uma delas. Sally Davies, repórter da FT Technology As palavras bloco e corrente são usadas separadamente no artigo original de Satoshi Nakamoto, para descrever a tecnologia e o processo subjacente - mas acabam sendo popularizadas como uma única palavra, "blockchain".

2014 2016	VOLATILIDADE GERA DESCONFIANÇA CRIMINALIDADE GERA IMAGEM PÉSSIMA PARA ALGUNS É UMA LINHA DE VIDA A hiperinflação venezuelana faz com que o preço de uma xícara de café suba 350.000%. Os venezuelanos recorrem ao BTC como uma moeda alternativa para comprar mercadorias do dia a dia.	NOVAS OPORTUNIDADES - Os empreendedores de tecnologia começam a perceber que a blockchain pode ser usada para mais do que transações de Bitcoin. LANÇAMENTO DO ETHEREUM - Vitalik Buterin lança Ethereum - a segunda blockchain pública - para registrar outros ativos, não apenas moeda, incluindo contratos inteligentes.
2017 2021	CHINA BANE "ICOS" (Initial Coin Offerings) E "EXCHANGE" - Promove uma queda instantânea de 6% no valor do Bitcoin e o fechamento das operações na China. CHINA PRESTES A LANÇAR SUA PRÓPRIA MOEDA DIGITAL - O Banco Popular da China (PBOC) parece prestes a se tornar o primeiro grande banco central a emitir uma versão digital de sua moeda. FACEBOOK ANUNCIA LIBRA O Facebook anuncia Libra, acelerando o pensamento sobre 'stablecoins' por reguladores em todo o mundo.	A COMUNIDADE DE NEGÓCIOS SE ENVOLVE - Empresas da Fortune 500, Startups de blockchain e grupos de pesquisa lançam a Enterprise Ethereum Alliance (EEA) SURGIMENTO DE CASOS DE USO - Proteger identidade digital. - Habilitar tokenização de ativos. -Gerenciamento de dados pessoais para proteger privacidade. - Estará conectado com dispositivos IoT. REGULAMENTO PIONEIRO - Cingapura, Japão, Suíça, Lituânia, França e outros começam a ser pioneiros em uma nova estrutura regulatória para apoiar a blockchain.

Tabela 1 - Histórico Bitcoin / Blockchain

FUTURO PARA BITCOIN	O FUTURO PARA BLOCKCHAIN
Apenas 21 milhões de Bitcoins serão lançados. Aproximadamente 85% foram lançados até agora. Desafios: As transações em escala de Bitcoin são lentas: 7 transações/segundo, em comparação com os 1000/segundo da Visa Regulamentação: Tanto os EUA quanto a UE indicaram que irão regular o Bitcoin. 'Baleias': Alguns investidores possuem Bitcoins suficientes para movimentar o mercado, causando risco as operações globais, podendo manipular o mercado. Mercado: Não há muitos lugares onde você pode gastar seu Bitcoin no mundo real, então este será um desafio. Hoje em 2022 – Mercado mais maduro, investidores institucionais entraram e a *Lightnetwork* dá escalabilidade ao Bitcoin	Em 2019, a blockchain entra no 'Vale da Decepção' (*Gartner Consulting)*. O sucesso virá quando a blockchain submergir mais ainda suportando os picos. Esses picos serão novos aplicativos e serviços que irão melhorar nossas vidas. Outros picos serão aprimorados e surgirão modelos e processos de negócios colaborativos que vão transformar negócios em todos setores. Muitas, até mesmo a maioria das inovações fundadas em blockchain, envolverão sinergias com outras novas tecnologias. Como a base do iceberg, uma indústria de blockchain madura será invisível para o usuário. Hoje em 2022 muito já se tornou realidade.

Tabela 2 - Futuro Bitcoin / Blockchain

3.3 O que é? Como funciona?

Como visto no *white paper* de Satoshi Nakamoto citado anteriormente, a *blockchain* é uma tecnologia relativamente recente, porém que implementa uma solução computacional dos anos 1990 originada nas principais aplicações disruptivas daquela década como a Napster e o Gnutella. A blockchain oferece suporte distribuído, confiável e seguro para realização de transações entre participantes que não precisam ter confiança entre si e que estão dispersos em larga escala numa rede P2P. É considerada disruptiva, pois cria uma entidade autenticadora descentralizada, eliminando a necessidade de uma terceira parte de confiança. Dessa forma, pode substituir entidades certificadoras centralizadoras tais como bancos, governos, etc.

Blockchain é resultado de uma engenhosa combinação de técnicas provenientes da computação distribuída confiável (tolerância a falhas, sistemas P2P, etc.), criptografia (chave assimétrica, funções *hash*, desafios criptográficos, etc.) e teoria dos jogos (mecanismos de incentivos). As redes P2Ps são soluções de conectividade ponto a ponto que não necessitam de um mecanismo de servidor centralizado para compartilhar informações diversas. Neste modelo de implementação as informações de endereçamento dos dados podem ser obtidas em tempo real na rede, direcionando o demandante para diversas máquinas distribuídas na *internet* que possuem o mesmo dado e podem, de forma distribuída, compartilha-lo dando robustez a rede e a capacidade de compartilhamento sem riscos de suspenção do serviço durante uso ou download da informação.

O processo de identificação do mesmo dado distribuído na internet pela rede P2P é realizado por uma função de *hash* [35] que, através dos aplicativos de conexão destas redes, direcionam e compartilham os arquivos multiplicados na *internet*. Como visto na Figura 3.1, cada arquivo, dado ou informação compartilhada em uma rede P2P é codificada por uma função de *hash* e produz um *hash* resultante, a partir de então, esta informação pode ser multiplicada, compartilhada e distribuída na internet.

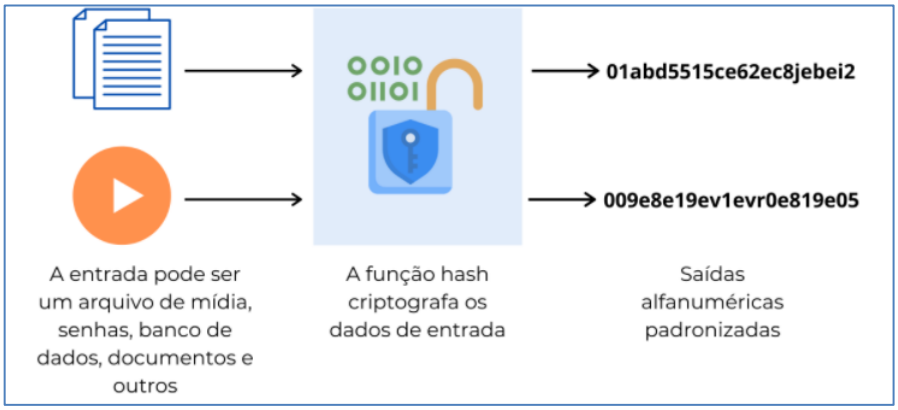

Figura 3.1 - Processo de criptografia da função *hash*.

Depois do momento de cifragem por *hash* e distribuição da informação pela internet, o dado pode ser recuperado conectando-se um usuário ou demandante pela informação a um arquivo de índice que possui a função de *hash* resultante. Através de aplicativos de busca na internet, os mesmos arquivos compartilhados podem ser localizados em diversas maquinas ou servidores e por fim, reconstruídos localmente por um processo de download cooperativo, ou seja, todos os nós (pontos da rede) que possuem o arquivo originalmente distribuído cooperam para recuperação do mesmo, devido a capacidade da função de *hash*, em identificar as cópias e seus blocos espalhados pelo ambiente virtual.

O funcionamento da *blockchain* é estruturado da seguinte forma [36]: cada nó conectado a rede *blockchain* possui um cópia do banco de dados original de transações daquela rede desde a primeira operação. Como se vê na Figura 3.2, cada transação nova é autenticada pelo *hash*, registrada e vinculada ao registro anterior, assim, se torna computacionalmente impossível uma terceira parte tentar alterar algum registro prévio ao bloco atual uma vez que, além de alterá-lo precisará reconstrui-lo em cada nó multiplicado na rede. Lembrando que cada registro de bloco na *blockchain* é criado em aproximadamente 10 minutos. Por fim o "passado" de um registro da *blockchain* é computacionalmente imutável, apenas o "presente" continua registrado e, sempre é possível ver toda a "história" passada.

O bloco inicial da blockchain que registra o estado inicial do banco de dados na sua gênese é seguido pelos blocos subsequentes, onde cada qual contém um grupo de transações já validadas. Como cada bloco subsequente contém um *hash* do bloco anterior, cria-se assim o encadeamento entre eles e garante-se a integridade da informação, como é impossível alterar blocos antigos sem alterar os blocos subsequentes, o que seria percebido pelos demais nós.

Uma *blockchain* pode ser compreendida como um estado inicial seguido de um certo número de transações agrupadas em bloco. De fato, o estado atual do banco de dados está contido em uma b*lockchain* apenas de maneira abstrata, sendo necessário que cada nó valide tal estado, partindo do inicial e aplicando as subsequentes autenticações de transações.

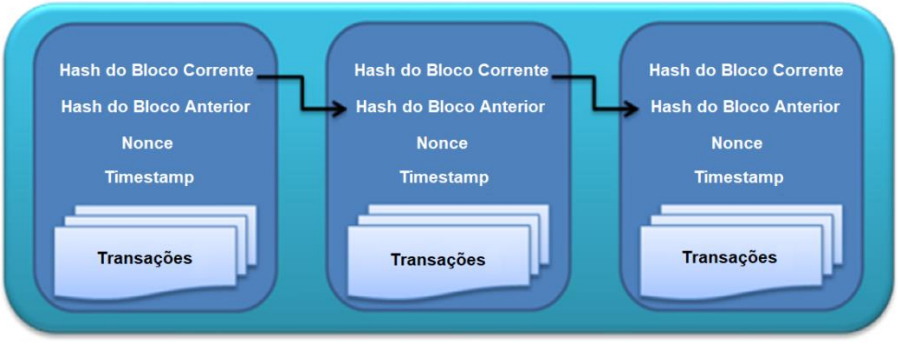

Figura 3.2 - Representação gráfica de uma blockchain.

Os blocos são criados através de um processo da teoria dos jogos onde cada nó da rede que faz a autenticação das transações, recebe uma compensação por este trabalho, sendo este processo conhecido como “mineração”. Um nó pega um certo número (geralmente delimitado pelo consenso da maioria) de transações já validadas e as inclui no bloco. Em seguida, deve criar o cabeçalho do bloco. Cada *blockchain* pode determinar diferentes informações que devem compor este cabeçalho, mas via de regra ele deve conter o *hash* do bloco anterior para criar o encadeamento e uma marca de tempo indicando quando foi criado. Uma vez concluído o bloco, obtém-se seu *hash* que é gravado juntamente com ele para identificá-lo e para ser incluído no cabeçalho do próximo bloco.

Outro passo comum na criação deste cabeçalho consiste em organizar as transações em uma Árvore de Merkle e gravar a raiz desta árvore no cabeçalho. Uma arvore de Merkle é uma estrutura de dados utilizada para verificação da integridades das informações em ambientes distribuídos, ou seja, em ambientes onde não há centralidade no processamento dos dados. Árvores de Merkle adotam a estrutura de uma árvore binária onde cada folha contém o *hash* de uma informação. Os nós superiores, ou pais, armazenam o *hash* da combinação dos *hashes* dos filhos aumentando em muito a força criptográfica da informação.

Isso permite que se faça um *hash* apenas do cabeçalho para servir como assinatura do bloco (já que a raiz de Merkle é um *hash* das transações) e permite a verificação de transações específicas usando uma “Prova de Merkle”. Na Figura 3.3 tem-se a árvore Merkle do Ethereum onde as transações estão nas folhas e apenas a raiz desta estrutura vai para o cabeçalho da *blockchain*, acelerando assim o processo de *hash*.

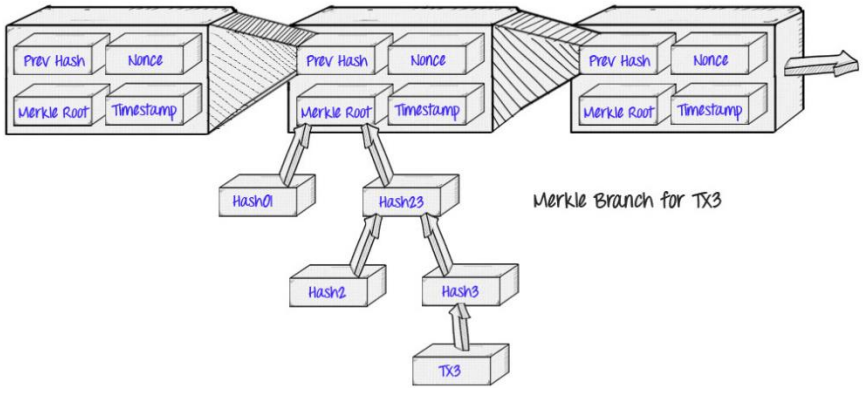

Figura 3.3 - “Merkling in Ethereum” by Vitalik Buterin

Com o propósito de se evitar que ocorram ataques ou fraudes ao sistema é necessário que exista algum mecanismo de consenso que incentive aos mineradores a serem bem-intencionados. Neste sentido, foi proposto pelo criador do sistema, como Bitcoin, Ethereum e outras moedas digitais, o que se chama de Prova de Trabalho (*Proof of Work* - PoW). O PoW

é uma função difícil de calcular, mas fácil de verificar. A função possui uma mensagem, um endereço de destinatário e alguns outros parâmetros.

Em 1999, no artigo "*Proofs of work and bread pudding protocols*" publicado por Markus Jakobsson e Ari Juels [37], o termo "*Proof of work*" foi introduzido e a notação foi criada. Seu objetivo era caracterizar a notação de uma prova de trabalho, em um protocolo no qual um provador demonstra a um verificador que ele gastou um certo nível de esforço computacional em um intervalo de tempo específico.

Mas como funciona? Os mineradores tentarão adivinhar um número aleatório que deve fornecer o *hash* certo para esse bloco de transações. Nesse processo, há duas coisas que se precisa definir: função *nonce* e *hash*.

Nonce – é um número aleatório usado apenas uma vez. No caso do Bitcoin, esse número é um número inteiro entre 0 e 4.294.967.296.

Hash – é um algoritmo ou fórmula que converte qualquer sequência de caracteres em uma sequência de 64 caracteres ou números.

Como se sabe, todos os blocos da *blockchain* têm seu *hash* (id). Essa é uma *string* que se atribui ao bloco quando o mesmo é criado. Portanto, primeiramente quando se deseja verificar e/ ou criar um novo bloco, pega-se o *hash* do bloco anterior e o adiciona com o bloco de transações do bloco em questão, produzindo no final um grande bloco de texto. Agora, com estas informações, pode-se iniciar os cálculos e validar o próximo bloco ou checar um bloco existente. Para os cálculos, aplica-se a função *hash* neste bloco de texto, incluindo-se também um número aleatório (*nonce*) modificado em cada interação de calculo, até que se obtenha uma sequência de bits coerente com certo número de zeros à sua frente. A quantidade de zeros é especificada pelo grau de dificuldade da rede *blockchain* naquele momento de calculo. Por fim, como se pode obter mais de um *hash* que atende a exigência de zeros, quem encontrar o primeiro *hash* válido é considerado como minerador do bloco e assim ganha a recompensa. No momento de validação/criação de um bloco, este novo *hash,* produzido no conjunto formado pelo *hash* anterior, as transações e o *nonce,* é incluído no cabeçalho do bloco seguinte juntamente com a informação de tempo de criação (*timestamp*) e outras. Desta forma também é possível verificar qualquer outro bloco já registrado na *blockchain* pelo calculo do *hash* referente ao bloco específico.

Parece fácil, mas o computador precisa executar cerca de 1021 cálculos, para encontrar o número certo (novo *hash*). Esse número não é pequeno e leva-se cerca de 10 minutos para encontrar o número certo que fornecerá a sequência necessária. Para uma função de *hash*, o Bitcoin usa o algoritmo de *hash* SHA-256.

Como o SHA256 é um algoritmo pseudoaleatório com resultado imprevisível, não há qualquer maneira de se obter uma "Prova de Trabalho" válida exceto por tentativa e erro alterando o *nonce* a cada tentativa. O valor do grau de dificuldade é recalculado a cada 2016 blocos, de modo que seja criado em média um bloco a cada dez minutos. Para incentivar os mineradores, o algoritmo adiciona uma transação ao bloco no qual se recompensa em frações da moeda digital o minerador, pela criação do bloco antes de iniciar a criação do cabeçalho. Assim, todo dia milhões de nós competem entre si na mineração de blocos a fim de ganhar esta recompensa e auditam uns aos outros para garantir que ninguém esteja "roubando".

Existem outros mecanismos de consenso possíveis como a Prova de Participação (*Proof of Stake* - PoS) e a Prova de Queima (*Proof of Burn* - PoB). Na primeira, os nós são selecionados de maneira aleatória para minerar um bloco, mas tem sua probabilidade aumentada quando atendem certos critérios (por exemplo, o critério de "idade da moeda" aumenta a chance do nó que possui uma certa unidade de valor a mais tempo). Já a prova de queima propõe que o criador do bloco "queime" (transfira) um pouco da sua criptomoeda para um endereço do qual não é possível recuperá-lo para provar que não está mal-intencionado.

A validação de um novo bloco segue um procedimento simples:

•Verifica-se que o bloco anterior referenciado existe e é válido;

•Verifica-se a marca temporal do novo bloco, que deve ser maior que a do anterior e não pode estar muito distante no futuro (diferentes blockchains determinam diferentes limites);

•Verifica-se quaisquer informações adicionais que cada blockchain específica coloca no seu cabeçalho;

•Verifica-se o mecanismo de consenso do novo bloco; dado o estado do banco de dados calculado até o bloco anterior e aplica-se (em ordem) todas as transações registradas no novo bloco a este estado, obtendo uma variedade de estados intermediários.

Se a aplicação de alguma transação resultar em um estado intermediário inválido, a verificação falhou; caso contrário, a verificação foi bem sucedida e o nó guarda para si o novo estado atual da blockchain, com o novo bloco. A validação deste novo bloco é feita pelo consenso da rede que está minerando este bloco, quando outros nós da rede chagam ao mesmo resultado calculado posteriormente ao primeiro. Esta validação é feita por cada nó quando recebe a informação de que um novo bloco foi criado. É por isso que o sistema funciona a base do consenso da maioria. Se em qualquer etapa da criação de um bloco o minerador fizer algo que vá contra o consenso, seu novo bloco não será aceito pelos demais, tendo então desperdiçado tempo e energia.

Por se tratar de uma rede distribuída, frequentemente acontece de dois ou mais blocos diferentes, todos legítimos, serem criados quase que simultaneamente e adicionados à mesma posição na Blockchain em nós diferentes da rede. Isso cria um *fork* temporário do banco de dados, no qual duas ou mais versões competem para serem aceitas. Cada rede deve ter definido no algoritmo algo que permita avaliar estas versões e decidir qual a vencedora. Por exemplo, em blockchains que usam PoW, a pontuação é calculada como o trabalho total acumulado em uma cadeia. Blocos criados que acabam ficando de fora da rede após este "desempate" são chamados blocos órfãos. Os blocos órfãos são descartados e não são replicados e aceitos nos nós da rede. Em geral, estes algoritmos são desenvolvidos de tal modo que, essa pontuação continue crescendo à medida que se adicionam novos blocos, de modo que a possibilidade de um certo bloco se tornar órfão diminui exponencialmente quanto mais a cadeia cresce.

3.4 Aplicações Atuais

A blockchain se caracteriza pela adoção do benefício de uso em aplicações diversas, o potencial de transformação é imenso e aplicações estão surgindo a partir desta tecnologia em inúmeros setores: finanças, saúde, artes, governo, além da própria computação (protocolos de redes, nuvem, IoT, etc.). Veja algumas delas [38];

Contratos Inteligentes: Contratos automatizados que são incorporados como um código *if-this-then-that* (Se-Isso-Então-Aquilo), que lhes dá auto execução. Na vida real, um intermediário garante que todas as partes sigam os termos. A *blockchain* não apenas renuncia à necessidade de terceiros, mas também garante que todos os participantes da rede conheçam os detalhes do contrato e que os termos contratuais sejam implementados automaticamente quando as condições forem atendidas. Pode-se usar contratos inteligentes para todos os tipos de situações, como derivativos financeiros, prêmios de seguro, leis de propriedade, acordos de financiamento coletivo, entre outros

Gerenciamento de Identidade: O ponto crucial das transações financeiras on-line exige gerenciamento de identidade. No comercio eletrônico (*web commerce*), as soluções para riscos de segurança não estão equacionadas. Para transações *on-line*, uma identidade segura é importante e é muito difícil de encontrar, dessa forma têm-se enormes oportunidades nesta área.

Proteção a Direitos de Propriedade Intelectual: O plágio e a exploração da propriedade intelectual são uma situação particularmente problemática no mundo digital. Conteúdo rico,

gratuito, pode ser reproduzido e replicado em toda a Internet. A *blockchain* pode ser uma grande ajuda nesse sentido, sabendo que a cópia e a redistribuição podem ser evitadas.

Mercado de Valores Mobiliários: O aumento da eficiência na liquidação de ações é uma das principais razões pelas quais as *blockchains* são ideais para o mercado de valores mobiliários. Atualmente, são necessários três dias desde a compra de uma garantia até a propriedade. Com a execução ponto a ponto, as confirmações de negociação podem ser feitas em tempo real e isso é um grande fator de mudança.

Pagamentos e Transferências Internacionais: O dinheiro vai de um banco para outro e esse processo continua até atingir as contas finais. O problema aqui é que cada banco tem seu próprio livro caixa, sendo necessário que haja reconciliação no final de cada dia, tornando o processo caro e muito lento. Se todos os bancos compartilhassem um registro único (livro-registro), estes poderiam se conectar a este registro e assim, agilizar a remessa do dinheiro.

Saúde: Prejudicado por interações complexas entre médicos, hospitais, seguradoras, farmacêuticos, pesquisadores e muitos outros participantes, o sistema de saúde precisa de melhores soluções para garantir acesso e compartilhamento de informações sensíveis. Nenhuma instituição pode solucionar o problema da confiança de forma definitiva, mas *blockchain* poderia ser usada como um mecanismo para controlar o acesso a registros médicos, e poderia fornecer uma trilha de auditoria para o acesso a esses registros.

Educação: Muitas instituições concedem certificados de vários tipos. Estes certificados variam desde diplomas universitários a certificações de especializações: Mas verificar se alguém realmente tem as certificações que afirma ter é difícil. Uma *blockchain* poderia ser usada como um repositório de credenciais e qualquer instituição poderia adicionar suas credenciais na *blockchain*, e qualquer um que precisasse, poderia verifica-las.

Trade Finance: Uma carta de crédito refere-se a um documento, com a precisão do tempo e do valor, dando um crédito a um comprador. Se um comprador não puder efetuar os pagamentos, o banco cobrirá o valor total ou restante da compra. O processo típico de emissão de carta de crédito é muito demorado e baseado em papel. Esse processo lento cria problemas de liquidez. Usando *blockchain*, isso pode ser feito em poucas horas.

Votação Digital: Em países com práticas ilícitas em mecanismos de votação, especialmente devido a governos corruptos, a *blockchain* poderia ser realmente útil. Quando várias partes estão envolvidas em cuidar do mecanismo de votação com *blockchain*, torna-se bastante inviável adulterar tal sistema.

Internet of Things: Há muitas coisas que uma *blockchain* pode fazer com IoT. De longe, o maior problema da Internet das Coisas é proteger dispositivos e atualizar o software.

Bilhões de dispositivos inteligentes vulneráveis na Internet são um problema de segurança de escala inimaginável. Quando um dispositivo recebe uma atualização, como ele sabe que a atualização é legítima? Uma *blockchain* pode ser usada de maneira padronizada para provedores de software e dispositivos, para provar sua identidade, registrar seu estado atual e validar a atualização como legítima. Essa á a principal temática deste trabalho.

3.5 As principais cadeias de blocos

Atualmente existem inúmeras representações digitais chamadas de *currency* (moeda). No jargão das criptmoedas são denominas também como “Altcoins” todas as demais moedas que surgiram posteriormente ao Bitcoin. As chamadas moedas digitais alternativas já compõem um grupo de mais de 4.000 criptomoedas [39]. Elas buscam trabalhar sobre algumas limitações técnicas do Bitcoin como escalabilidade, velocidade de transação, entre outras, criando assim tecnologias mais avançadas e com maiores vantagens competitivas.

Mesmo com as Altcoins utilizando diversos recursos já existentes como base para o desenvolvimento de novas funções e recursos, essas criptomoedas alternativas variam muito entre si. Do ponto de vista financeiro ou de investimento [40], quando se olha uma criptomoeda, em primeiro lugar deve-se avaliar se o objetivo é especular ou investir. O mercado dos criptoativos permite que se faça as duas coisas.

Para fazer *Trade* (operações de compra e venda), moedas mais líquidas como Bitcoin, Ethereum, entre outras são mais estáveis e possuem rede de liquidez maior. Já no caso de investimento e risco, as altcoins, moedas de menor valor nominal são as que têm grande potencial de valorização.

É imprescindível antes de se escolher uma *blockchain* para se utilizar em um projeto, que seja avaliada a sua relevância de mercado. Uma vez que a solução poderá estar ancorada em uma blockchain desestruturada corre-se o risco de apostar em uma solução que poderá ser extinta em um futuro breve. Todo o processo de escolha de uma blockchain passa necessariamente pela sua força econômica. Para entender quais as moedas e por consequente quais blockchains podem ser mais promissoras, deve-se analisar alguns pontos importantes:

- Projeto - Para escolher os melhores projetos para investir, deve-se conhecer o projeto da criptomoeda. Claro que isso nem sempre é simples de ser feito, pois muitos deles são técnicos.
- *Marketcap* - A análise precisa estar sempre relacionada a dois conjuntos de informações: posição no ranking e preço. Para que se possa analisar o *marketcap* das moedas,

é necessário consultar paginas de cotação (*exchange*) que apresentam o valor de comercialização daquela moeda e sua precificação comparativa junto ao Bitcoin.

- Posição no Ranking
  - ✓ Absoluta: A posição em relação às demais criptomoedas listadas em sites específicos de operação monetária. Essa informação ajuda a identificar quais de fato estão fortes dentro do cenário completo.
  - ✓ Relativa: Nessa forma de análise se procura estabelecer, dentro da amostra que foi extraída da posição absoluta, quais estão mais bem classificadas dentro dos critérios estabelecidos para volume de mercado.
- Valor nominal baixo - Como pode-se ver na Figura 3.4, dentro dos critérios utilizados, temos uma relação de 5 criptomoedas com preços bem atrativos. Assim conseguimos uma oportunidade de definição em qual blockchain apostar a nossa solução.

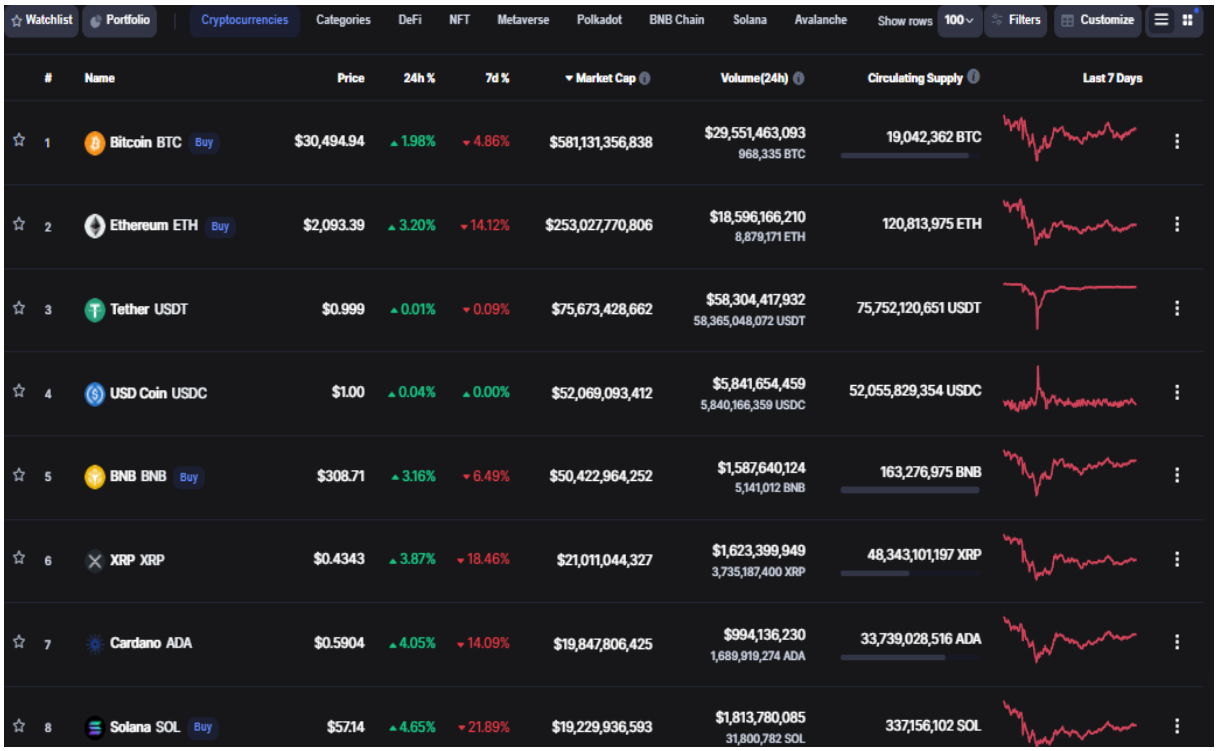

Figura 3.4 - Preços de criptomoedas pelo site www.coinmarketcap.com (17/05/2022).

Vale lembrar que o mercado de criptomoedas cresce a cada dia e mais e mais pessoas estão aderindo a essa nova forma de investir seu dinheiro. Sabemos que esse mercado ainda está longe da maturidade e por isso os riscos são elevados. É necessário avaliar constantemente as moedas digitais e consequentemente as suas *blockchains*, pois a atenção na hora de escolher onde implantar a solução será o diferencial entre a sobrevivência ou não de uma plataforma.

3.6 Ethereum

Ethereum [41] refere-se a uma plataforma de *software* de código aberto baseada na tecnologia *blockchain*, que permite que os desenvolvedores criem aplicativos descentralizados que também são chamados de dApps. No entanto, a palavra Ethereum também é utilizada para se referir à moeda Ether (ETH), uma criptomoeda criada na plataforma Ethereum.

A história do Ethereum começa com Vitalik Buterin, em 2011. Buterin tomou conhecimento das deficiências do Bitcoin e criou o Ethereum como uma tecnologia de *blockchain* superior. Ele achava que aqueles que fazem parte da comunidade Bitcoin não estavam enfrentando os problemas de forma adequada. Eles estavam desenvolvendo aplicativos individuais, que tentavam dar um suporte explícito a cada caso de uso possível da tecnologia *blockchain*, em um tipo de protocolo de denominado “Canivete Suíço” (Tentar fazer tudo com o Bitcoin).

Segundo [42], Buterin em seu discurso de 2014, explicou que o Ethereum seria uma plataforma que trabalharia no conceito de contratos inteligentes. Com essa abordagem, o fundador do Ethereum apresentou uma lista de descrições, como por exemplo, a oportunidade de se registrar e executar em sua blockchain ações automáticas de execução de contratos, que antecipavam diversas inovações no ecossistema de criptomoedas. Entre as ferramentas que se tornaram populares na rede, muitas das quais já existiam no Bitcoin, são os aplicativos descentralizados, *tokens* não dispensáveis e finanças descentralizadas (DeFi).

Contratos inteligentes - Para a implementação desses contratos, os criadores do Ethereum introduziram a linguagem de programação *Solidity* [43]. Uma linguagem de Turing completa que facilita a programação de um computador para executar uma variedade de operações. Desta forma, a linguagem permite que qualquer pessoa crie contratos inteligentes na *blockchain*. Para isso, só precisa escrever a lógica em algumas linhas de código.

Finanças Descentralizadas (DeFi) - O mercado de DeFi é o grande impulsionador no ecossistema de criptomoedas atualmente. Ele surge como uma alternativa ao financiamento tradicional e aos serviços e produtos financeiros oferecidos pelos bancos. DeFis são definidos como um novo ecossistema financeiro, descentralizado, global, transparente, resistente à censura, sem intermediários e de fácil acesso, onde cada usuário tem controle de seus ativos.

Aplicativos descentralizados (dApps) - São trechos de código escritos em contratos inteligentes. Os dApps se comunicam com a *blockchain* e são programados para controlar várias ações. Eles processam, por exemplo, as informações externas que recebem, enquanto os códigos são executados em uma rede P2P. O funcionamento de um dApp depende de dois

elementos: uma rede como o Ethereum e um ambiente de execução. A *blockchain* permite que o aplicativo tenha uma infraestrutura de rede descentralizada.

O Ethereum será atualizado (2.0) no futuro próximo para o Ethereum 2.0 planejada para o ano de 2022/23. O recurso principal é uma alteração de uma validação de prova de trabalho para prova de participação. O Ethereum 1.0 é a tentativa de construir o computador mundial, já o Ethereum 2.0 (com PoS) será, de fato, o computador mundial, ou seja uma maquina computacional distribuída em todos os nós de validação da rede, capaz de processar e devolver informações. Um protocolo de prova de participação significará que os usuários "participam" com seus ETH como garantia para a verificação de uma transação (e reivindica a recompensa).

O valor atual do Ethereum é de algo em torno de US$2.093 (17/05/2022) e ele continua mantendo o segundo lugar para Bitcoin no mercado de negociação de criptomoedas.

3.7 Considerações sobre *Blockchains*.

É importante ressaltar que a tecnologia *blockchain* da rede Ethereum que possibilita a implementação de contratos inteligentes não é a única disponível hoje. Existem várias outras redes blockchain criadas recentemente que propõem soluções idênticas e até melhores que o Ethereum, mas ainda não atingiram uma maturidade crítica que permita a operação segura de uma plataforma cujo objetivo principal é a autenticação e segurança para circuitos integrados e dispositivos.

É importante notar que toda rede descentralizada criada após o Bitcoin veio para tentar resolver o trilema da *blockchain* que se baseia em três aspectos fundamentais: Segurança, Descentralização e Escalabilidade. O Bitcoin atualmente possui o maior poder computacional entre as redes, mas a quantidade de transações realizadas por segundo (escalabilidade) especificada no protocolo não é suficiente para que seja utilizado como meio de pagamento em tempo real. Quanto à segurança (força contra ataques à rede) e descentralização (capacidade de manter a autenticação de transações o mais distribuída possível), não há outra *blockchain* que faça frente à sua capacidade.

Com a intenção de apresentar uma discussão prévia sobre qual rede poderia atender melhor a nossa proposta de forma eficiente, é preciso primeiro entender que só é possível implementar contratos inteligentes em redes *blockchain* que permitam tal recurso, reduzindo drasticamente o número de protocolos que poderíamos usar. Um exemplo seria a exclusão do próprio Bitcoin por não implementar tais contratos em seu blockchain.

Dentro desses protocolos, a segunda opção que devemos considerar é o quão seguro esse protocolo é, afinal em nosso trabalho o foco é garantir a autenticação, ou seja, deve ser preferida a escolha de uma rede com baixa capacidade de sofrer ataques e com maior histórico de resolução de vulnerabilidades. Por outro lado, deve-se considerar o fator de descentralização, que suporta grandemente a capacidade de *up-time* e a resiliência operacional da rede. Por fim, a característica relacionada à escalabilidade, embora importante, pode não ser prioridade neste momento. Tendo em vista que, pela característica existencial de um trilema, sempre atenderemos a dois requisitos, em detrimento de um terceiro pela própria ambiguidade do trilema.

Para avaliar quais são as melhores soluções para redes blockchain que implementam contratos inteligentes, observe a Figura 5, que apresenta uma lista das principais redes e suas funcionalidades. Eles são classificados de acordo com o trilema das redes blockchain.

COMPARAÇÃO DE PLATAFORMAS DE CONTRATO INTELIGENTE (PRINCIPAIS)						
	ETHEREUM	CARDANO	BINANCE	POLKADOT	SOLANA	FANTOM
Arquitetura	Single-chain (synchronous)	Single-chain	Single-chain (synchronous)	Mulchain (parachains)	Single-chain (synchronous)	Single-chain (synchronous)
Sgurança	Global	Blockchain-specific	Shared	Shared (if parachain connected)	Global	Global
Concenso	Proof-of-Work	Proof-of-Stake	Proof-of-Stake (Autority)	Nominated Proof-of-Stake	Proof-of-History	Proof-of-Stake
VM/Linguagem	EVM (Solidity, Vyper)	K(EVM)	EVM (Solidity, Vyper)	WebAssembly, Substrate	Sealevel (Rust)	EVM (Solidity, Vyper)
Validadores/ Mineradores	301250	2076	21	297	1044	44
Taxas	Variable transaction fees	Variable transaction fees	Variable transaction fees	Market cost for parachain slot	Variable transaction fees	Variable transaction fees
Governança	Off-chain	On-chain (not yet)	On-chain	On-chain	On-chain	On-chain
Ecosistema	5000 + projects	200+ Projetcts	857 + projects	495 + projects	338 projects	140+ projects
Captalização de Mercado	397B	76B	63B	$38,7 B	47,7B	31B
Inico da Rede	jul/15	jul/20	ago/20	ago/20	mar/20	ago/20

Melhor Média Pior

Figura 3.5 - Blockchain - Comparação de Plataformas.

Este trabalho agora consegue identificar a melhor proposta em relação à rede blockchain ideal para se usar, e assim tem uma direção importante.

3.8 Novas possibilidades de Autenticação/Certificação de CI

Com a Internet das Coisas, ao mesmo tempo em que temos a possibilidade de desenvolver uma gama de aplicações interessantes (cidades inteligentes, saúde, casas inteligentes, etc.) há novos desafios, sobretudo com relação a segurança e privacidade. Nesse sentido, o casamento da IoT com a *blockchain* pode ser um diferencial por fornecer uma nova camada computacional para compartilhamento e análise segura dos dados, com maiores garantias de privacidade e segurança. Desta forma, essa camada pode ser utilizada para autenticar, autorizar, controlar e auditar as informações geradas por estes objetos inteligentes.

Atualmente ainda não se tem nada estabelecido no mercado de soluções de uso de certificação e autenticação para circuitos integrados ou dispositivos que façam uso de uma *blockchain* com contratos inteligentes. Mesmo sem a implementação interna de soluções de criptografia, a possibilidade de realizar o registro e posterior consulta sobre a autenticidade de origem de fabricação de um CI é um processo plenamente viável, porém este trabalho avança um pouco mais.

Deste modo, gerando um código randomizado com maior probabilidade de unicidade possível (chave criptográfica), pode-se utilizar a *blockchain* para registar, certificar e garantir, com altíssimo grau de confiabilidade a autenticidade de um componente semicondutor. Assim, pode-se tratar a autenticação de qualquer dispositivo de *hardware*, tanto na rastreabilidade de fabricação de seus componentes, quanto na consolidação de um novo dispositivo registrado e validado na *blockchain*. Esta é a proposta e seu modelo será apresentado no próximo capítulo.

4 Autenticação em Silício – O Modelo

Segurança digital e ambientes confiáveis não são algo que pode ser entregue por qualquer entidade. À medida que os formuladores de políticas adotam os semicondutores como a base para construir a infraestrutura do futuro e impulsionar a transformação digital, a indústria e o Setor Publico devem redobrar seus esforços em colaboração e comunicação. Desta forma este modelo apoia tecnicamente uma solução para autenticação mais simples e fácil de ser implementada.

4.1 Paradigma da Autenticação

O processo de autenticação fornece garantia sobre a identidade de um usuário. Pode-se chamar de "reclamante" o indivíduo (ou sistema) cuja identidade será verificada. Credenciais são a evidência que um reclamante apresenta para estabelecer sua identidade. Assim, o sistema de autenticação fornece proteção contra ataques externos como, por exemplo, *man-in-the-middle* (MITM), *spoofing* e outros.

O conceito por trás do ataque MITM é bastante simples e não se restringe ao universo *online*. O invasor se posiciona entre duas partes que tentam comunicar-se, intercepta mensagens enviadas e depois se passa por uma das partes envolvidas. Apesar de basear-se na mesma ideia, o invasor deve permanecer inadvertido entre a vítima e uma instituição verdadeira para que o golpe tenha sucesso.

Já o ataque de *spoofing* é uma falsificação de IP (protocolo de internet). Ou seja, ele falsifica a comunicação entre os dispositivos fingindo ser uma fonte confiável.

Tipos de Autenticação:

- Entidade: um usuário afirma ter uma identidade legítima em um sistema;
- Origem de dados: Fornece evidência de que dados, como uma mensagem de e-mail, foram originados de um usuário legítimo.

A autenticação de entidade pode ainda se subdividir em:

- Unilateral: Apenas uma das partes envolvidas na comunicação se autentica;
- Mútua: Ambas as partes devem se autenticar.

Dentro da ideia de autenticação pode-se apontar como paradigmas a serem tratados os seguintes pontos para tentar validar seu acesso [44]:

Algo que se sabe: o reclamante conhece a informação, como uma senha ou um PIN;

Algo que se possui: o reclamante demonstra a posse de algo, como uma chave física,

um crachá, um cartão ou uma chave privada em um *smartcard*;

Algo que identifique: baseado em alguma característica imutável do reclamante, como impressão digital, voz, padrão de retina ou geometria das mãos.

4.1.1 Algo que se Sabe – Senhas e PINs

Senhas são o mecanismo de autenticação mais usado para validar um usuário. Neste modelo, o reclamante prova sua identidade apresentando o conhecimento de uma *string* de caracteres. As senhas são uma das maiores vulnerabilidades em um sistema de autenticação, pois as pessoas tendem a escolher senhas fáceis de memorizar e por isso, fáceis de adivinhar.

Tipos de ataques a senhas:

- Pode ser descoberta se for transmitida sem criptografia em uma rede;
- Um intruso entra em um sistema e lê o arquivo de senhas;
- Alguém pode adivinhar facilmente uma senha mal escolhida;
- Pode ser possível quebrar uma senha usando um ataque de dicionário;
- É possível inclusive enganar um usuário fazendo-o revelar sua senha.

4.1.2 Algo que se Possui – Tokens

Um reclamante pode fornecer evidência de sua identidade ao demonstrar a posse de um token. Como apresentado na Figura 4.1, um *token* é um objeto físico, como uma chave, um documento de identidade, um dispositivo eletrônico ou um *smartcard*, por exemplo. Geralmente os tokens são usados em conjunto com uma senha para fornecer um grau mais elevado de certeza com relação à identidade de um reclamante.

Figura 4.1 - *Tokens* e *Smartcards*.

Smartcards - É um token do tamanho de um cartão de crédito que contém um microprocessador, memória para armazenar programas e dados, possuindo contatos elétricos usados como interface para um leitor de cartões, o qual também fornece energia ao dispositivo. Geralmente o *smartcard* possui um valor secreto armazenado, como uma chave privada de assinatura digital ou um número, usado para realizar a autenticação do portador do cartão.

Tokens Desconectados - É um tipo de *token* que não possui conexão física nem lógica com o computador cliente. Possuem um display que mostra um código de autenticação que muda, por exemplo, a cada minuto. O código pode ser usado então para realizar autenticação em um sistema, geralmente on-line, como um sistema bancário on-line (*bankline*).

4.1.3 Algo que Identifique – Biometria

A biometria usa características humanas físicas, comportamentais e morfológicas mensuráveis, com visto na Figura 4.2 para fornecer garantia da identidade de um reclamante. A biometria é usada geralmente em combinação com outros sistemas para obter-se um nível mais alto de segurança. Exemplos de Biometria:

Padrões de Retina – Testa padrões de vasos sanguíneos no tecido da retina da pessoa.

Impressões Digitais – Padrões únicos de impressão digital do usuário.

Formato da mão – Exame das medidas geométricas da mão de uma pessoa.

Padrões de Voz – O sistema explora os padrões vocais, acústicos, fonéticos ou até mesmo linguísticos.

Assinatura – Padrões únicos de uma assinatura manual convencional.

Figura 4.2 - A biometria e suas varias técnicas [45].

Por fim, a ideia por trás deste trabalho é explorar os principais paradigmas da autenticação e tentar construir uma proposta de modelo mais seguro contra ataques e fraudes cibernéticas, além de tentar garantir que o dispositivo “do outro lado da linha” é quem diz ser.

Neste modelo, trabalhou-se com o paradigma “algo que se sabe” através de senhas e PINs, quando se gera o identificador único do componente semicondutor e o transforma em uma chave segura para autenticação. Foi feito uso também do paradigma “algo que se tem”, uma vez que a informação de autenticação é publicada na *blockchain*, e vinculada exclusivamente a um dispositivo válido. Por fim o paradigma “algo que identifica” fica evidente quando se utilizada a extração de funções físicas não clonáveis (PUF) para produzir a unicidade e característica exclusiva do componente semicondutor.

4.2 PUF Utilizada

Com a intenção de agilidade no processo de demonstração dessa proposta, será empregada uma solução prévia já desenvolvida por outros pesquisadores [46] que implementam uma PUF baseada em osciladores em anéis ou *ring oscillators* (ROs) em circuitos de logica programável (*Field Programmable Gate Array* - FPGA). A implementação deste tipo de PUF, proposta orginalmente em [47], esclarece que tais estruturas são um tipo de PUF baseada em atraso, pois o valor de suas saídas varia de acordo com o atraso do circuito como um todo. Chama-se de circuito de atraso, o circuito que através de estruturas de repetição e loop (oscilador) produz um novo sinal de saída inverso ao da saída anterior. Essa inversão ocorre após um certo período de tempo (atraso) com relação à entrada, tendo a entrada realimentada por esta saída. Isso faz com que cada oscilador varie em uma frequência igual ao inverso do atraso total do circuito.

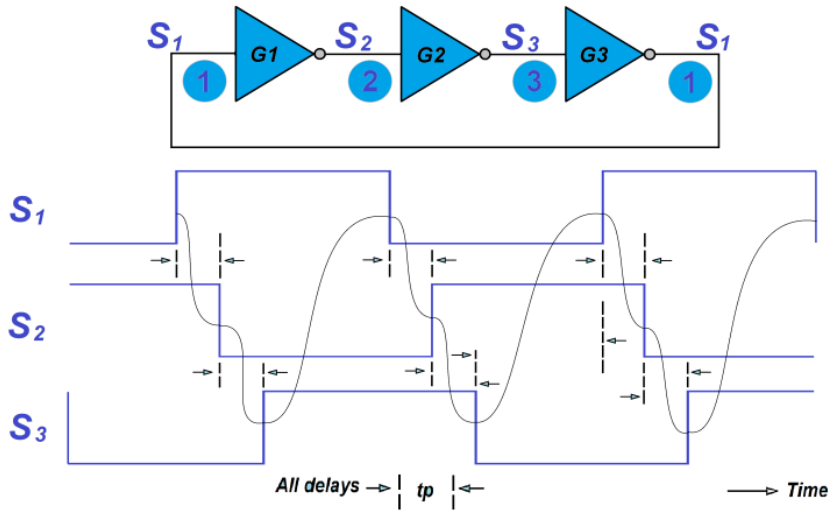

Figura 4.3 - (a)Oscilador em anel com 3 inversores; (b)Sinal de saída das portas lógicas[48]

Fator importantíssimo dessa implementação é a variação nas frequências que os osciladores podem sofrer naturalmente, principalmente devido à variações de tensão de alimentação, temperatura ou outros parâmetros externos. Assim, se faz necessário implementar formas de se compensar esse inconveniente e uma das formas é a utilização de uma divisão entre valores de frequência obtidos [46]. No modelo apresentado aqui neste trabalho, dois circuitos de atraso são escolhidos para serem usados no oscilador. Suas frequências são amostradas e o resultado é a razão entre as duas frequências. Nos casos em que a frequência dos osciladores tende a variar de forma linear com a temperatura e/ou tensão de alimentação, temos uma condição muito útil, fazendo com que a razão entre duas frequências produza um valor fixo. De acordo com [46], e conforme o Capitulo 2 deste trabalho, o valor obtido para as distâncias inter-classe e intra-classe com esses métodos são $\mu inter$=10x10^{-3} e $\mu intra$=0,1x10^{-3} [49].

Um método simplificado apresentado por [50] é baseado na comparação de frequências de dois osciladores diferentes. Nessa implementação, os osciladores da PUF são divididos em 2 grupos e a frequência de dois osciladores, um de cada grupo é amostrada, em seguida comparada de acordo com um teste lógico que pode ser levar a saída ao valor “0” caso a frequência do oscilador A seja maior do que a do oscilador B e levar a saída ao valor “1” em situação contrária. Com esse mecanismo juntamente com o método chamado de "*1-out-of-8 masking*", onde em cada 8 comparações apenas uma amostra e utilizada na saída da PUF, melhora-se a estabilidade do processo. Segundo [51], foram obtidas distâncias intra e inter-classe de 0.48% e 46.15% respectivamente. A Figura 4.4 exemplifica estes circuitos.

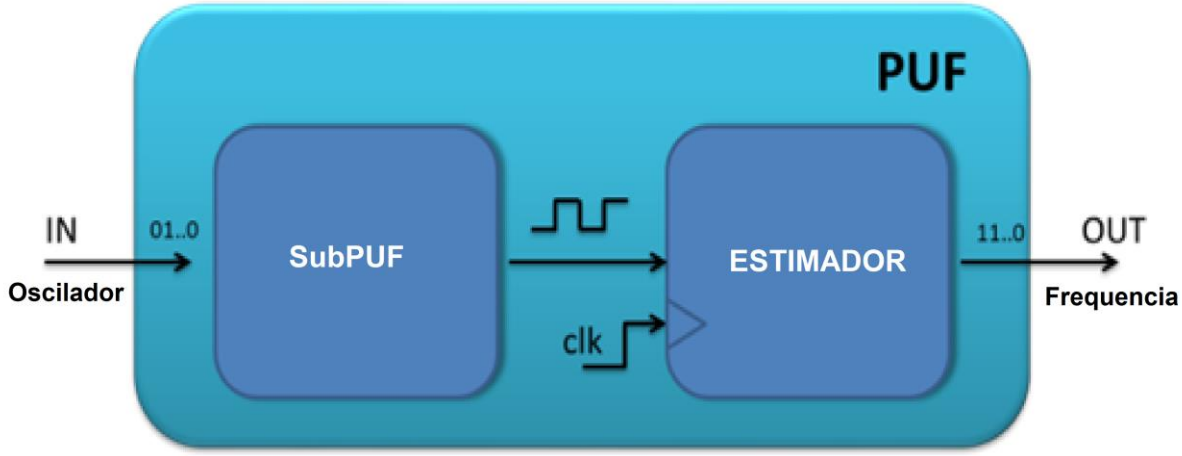

Figura 4.4 - (a) Compensação por Divisão, (b) Compensação por Comparação.

4.2.1 Detalhamento Funcional

Seguindo [46], a implementação da PUF baseada em osciladores em anel foi feita

utilizando-se um dispositivo de lógica programável FPGA, onde suas conexões internas e suas “tabelas verdade” são reconfiguráveis. Assim, pode-se implementar hardware de diferentes formas e testar diversas soluções diferentes.

O FPGA utilizada foi a Cyclone II da Altera, embarcada na placa Cyclone II DSP Development Board. Nessa placa, uma entrada JTAG é conectada em uma porta USB do computador e esta conexão permite a configuração da FPGA pelo computador. Para se fazer a comunicação com a placa de desenvolvimento, utiliza-se o software Altera Quartus II Web Edition vs 13.0 sp1. O Quartus II permite realizar a comunicação e a programação do FPGA, além de módulos adicionais como o SignalTap II, que permite o mapeamento de um analisador lógico dentro da FPGA, e por fim o aplicativo Chip Planner, que permite visualizar as regiões do FPGA que foram utilizadas na lógica programada e permite movimentar alguns blocos lógicos manualmente, caso seja necessário

Uma vez o FPGA programado, pode-se em tempo real de execução, acompanhar o fluxo da informação e os estados lógicos da FPGA recebidos de volta. Nesse fluxo de bits, os dados gerados pela PUF são capturados pelo analisador lógico implementado dentro da FPGA pelo SignalTap II, e assim os envia pela interface JTAG para a entrada USB do computador, onde podem ser processados.

4.2.2 Modelo de Medição Aplicada

Seguindo [46], a medição das saídas da PUF foi feita seguindo o fluxo:

- O analisador lógico espera que uma determinada condição de gatilho seja atingida de forma que a PUF possa ser reiniciada;
- O primeiro oscilador é selecionado;
- A frequência desse oscilador é medida ’n’ vezes. No caso em questão, foram escolhidas 42 medições por oscilador, devido a razoável estabilização do sinal observado;
- Após o mesmo oscilador ter sido medido ’n’ vezes, o próximo oscilador é selecionado e também é medido ’n’ vezes;
- Esse processo se repete até que todos os “m” conjuntos osciladores tenham suas frequências medidas;
- Os dados são armazenados em uma variável no sistema de aquisição de informações, ou seja, cada medição de frequência, independente do oscilador, é adicionada em um vetor numérico.

Conforme pode ser visto na Figura 4.5, os dois quadrados externos, representam um laço de repetição. Dentro do quadrado menor, tem-se uma representação de um oscilador (círculo com um ciclo de senoide) e de um medidor de frequência. As letras ’L’ e ’K’ no canto inferior direito do oscilador e do medidor de frequência mostram a qual laço de repetição cada componente está associado. Assim, ’n’ medidas de frequência serão feitas para cada um dos ’m’ conjuntos oscilador/medidor, e cada uma delas é armazenada em um vetor numérico.

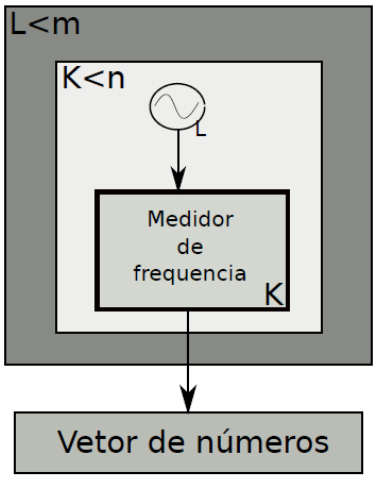

Figura 4.5 - Esquemático da sequência de medição [46].

Esse processo é denominado de sequência de medição assim, quando é dito que foi feita uma sequência de medição em uma PUF, o que se quer dizer é que a frequência de todos os osciladores foram medidas de acordo com o procedimento descrito.

4.2.3 Modelo de Análise dos dados

De acordo [46], diversas sequências de medição foram feitas com PUFs de configuração diferentes e os dados obtidos foram processados utilizando-se um código em Python. Esse código trata o seguinte procedimento para cada sequência de medição:

- Lê o arquivo no qual os dados foram gravados (vetor numérico);
- Separa os dados correspondentes para cada oscilador, ou seja, informa qual medida de frequência corresponde a qual oscilador;
- Calcula a média e o desvio padrão da frequência de cada oscilador nessa sequência;
- Compara-se o valor das médias das frequências para os mesmos osciladores em sequências de medição diferentes.

Assim, pode-se verificar o quanto a medição de frequência de um oscilador varia em

medições diferentes, ver o quão diferente são as médias das frequências de vários osciladores implementados entre si e em seguida, ver o quanto as médias das frequências dos osciladores variam para sequências de medição diferentes.

4.2.4 Implementação Física

A PUF implementada contém um banco de 256 osciladores (L) que são habilitados conforme a necessidade de medição. Não se fará uso nesta implementação de circuitos para compensação por comparação de duas frequências, nem está se levando em conta os fatores de comportamento transitório no momento de acionamento de cada oscilador, uma vez que a medição precisa da frequência não é importante agora. O que se necessita neste momento é uma implementação simples que mantenha a garantia de unicidade da informação para validação deste modelo. Estes estudos das variações para os osciladores serão deixados para trabalhos futuros. Cada oscilador é composto por uma quantidade impar de portas lógicas e por um circuito XOR de habilitação conforme indicado na Figura 4.6.

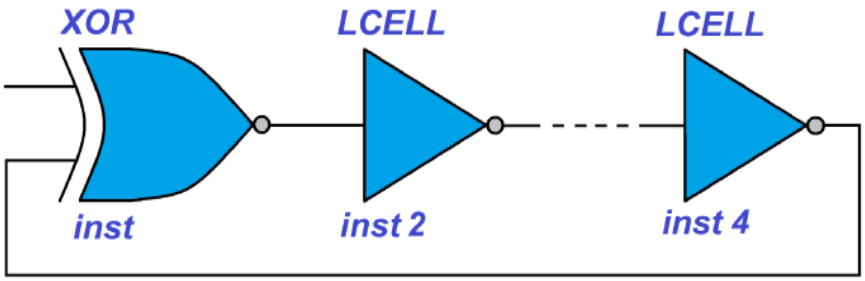

Figura 4.6 - Oscilador Implementado [46].

A implementação da PUF em si, será dividida em duas componentes: uma chamada subPUF e uma outra chamada de estimador, conforme Figura 4.7. A subPUF é composta pelos bancos de osciladores e um circuito capaz de selecionar e conduzir para a saída a frequência medida pelo no oscilador desejado. Esse oscilador habilitado, envia seu sinal para a saída da subPUF, que então o leva para a entrada de um próximo circuito denominado estimador.

O estimador, por sua vez, recebe o sinal do oscilador da subPUF e avalia a sua frequência em relação ao *clock* interno da FPGA. Essa frequência é então levada à sua saída, que por sua vez é enviada à saída da PUF.

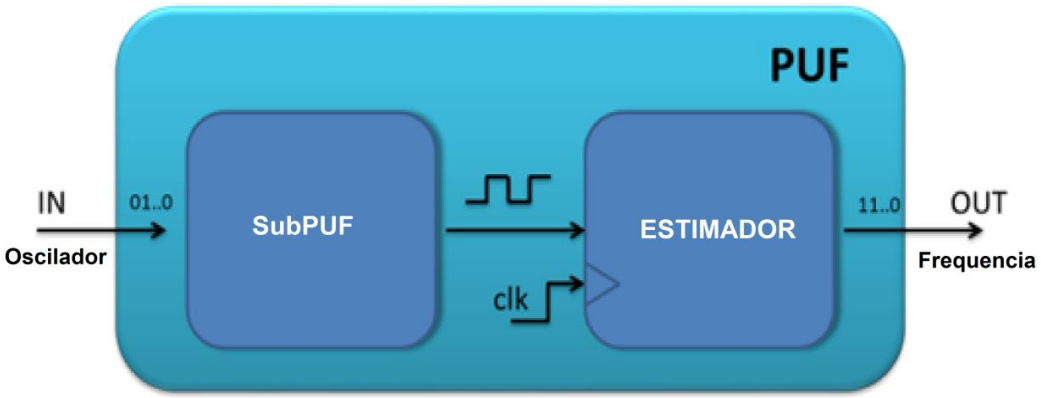

Figura 4.7 - Diagrama esquemático da PUF implementada [46].

A subPUF consiste em um decodificador, um multiplexador e um banco de osciladores. O sinal de *enable* de cada RO é conectado em uma saída do decodificador e cada saída do RO é conectada a uma entrada do multiplexador, assim os sinais de seleção do decodificador e do multiplexador são conectados no mesmo nó. Dessa forma, o mesmo oscilador habilitado pelo decodificador será selecionado pelo multiplexador. Um diagrama esquemático desse circuito pode ser observado na Figura 4.8.

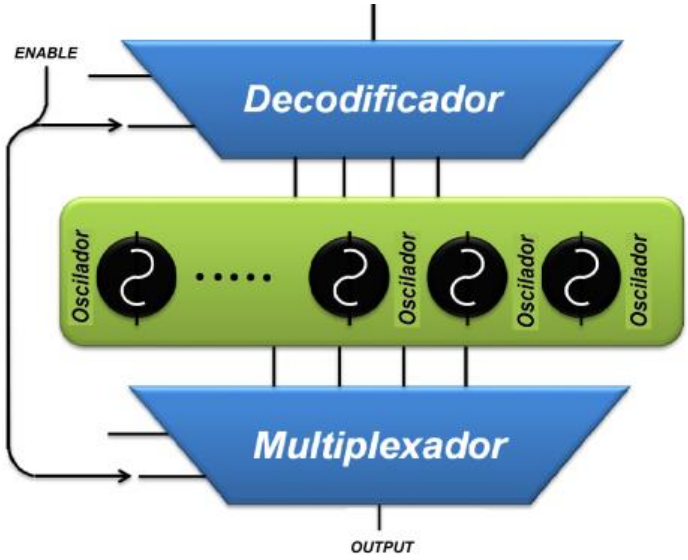

Figura 4.8 - Diagrama esquemático da subPUF [46].

Dessa forma, para o circuito descrito, dado um desafio (seleção do oscilador), tem-se uma resposta (frequência do oscilador). Os parâmetros que regulam a frequência dos osciladores, porém, variam de acordo com os componentes internos do oscilador e variam conforme a FPGA. Dessa forma, para o mesmo projeto de osciladores, espera-se ter osciladores diferentes em FPGAs diferentes. Esse tipo de comportamento onde dispositivos geram saídas diferentes dado o mesmo projeto é promissor na fabricação de PUFs.

4.2.5 Resultado de Eficiência

Os valores obtidos para as frequências segundo [46], foram calculados da seguinte forma: foram iniciados dois contadores simultaneamente, sendo um ativado pela borda de subida do *clock* interno e outro ativado pela borda de subida do sinal a ter a sua frequência medida. Quando o contador do *clock* interno atinge um valor pré-determinado, ele dispara um sinal que interrompe o contador ligado ao sinal medido. Desta forma, a frequência do oscilador será o valor do contador do sinal a ser medido dividindo-se pelo valor pré-determinado de ciclos de *clock* usados na medição. Tudo isso é multiplicado pela frequência do *clock* interno.

A primeira questão a ser investigada é o quão diferentes são as medidas de um mesmo oscilador em instantes de tempo diferentes. Para isso, realizaram-se várias sequências de medição e se coletaram os dados referentes a um só oscilador. Como cada sequência de medição captura 42 medições de frequência (amostras) para todos os 256 osciladores de uma vez, após as sequências de medição, apenas os dados de um oscilador com 5 células de atraso é armazenado. Na Figura 4.9 a seguir, tem-se a frequência de um mesmo oscilador em 6 sequências de testes diferentes. Desta forma, uma vez que cada sequência de medição captura 42 amostras de frequência de cada oscilador, cada curva da figura possui 42 amostras.

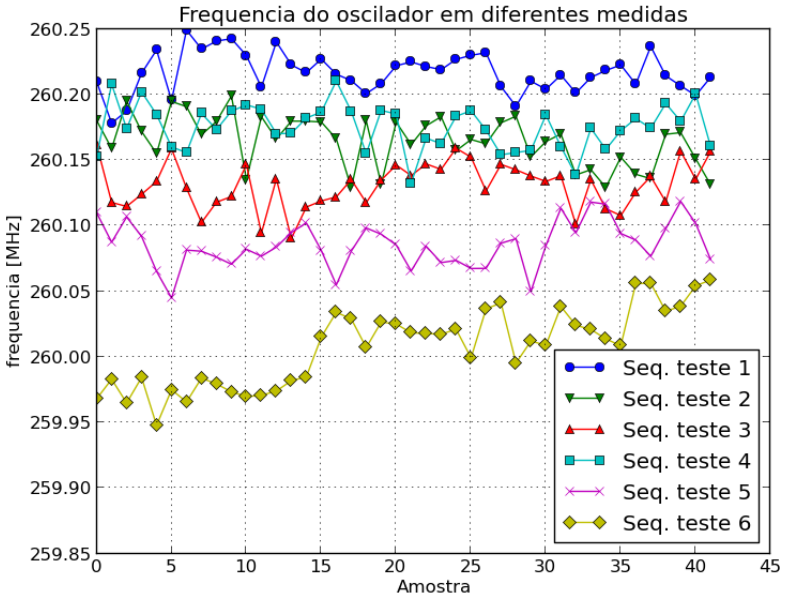

Figura 4.9 - Distribuição de frequência para um oscilador com 5 células de atraso em diferentes sequências de medição [46].

Uma vez que as medidas são independentes entre si e medem a frequência de um

mesmo oscilador, espera-se algum tipo de distribuição independente da sequência de teste. Porém, pode ser observado uma tendência onde as medições feitas na mesma sequência de medição são mais próximas entre si do que as medições de frequências em diferentes sequências de medição. Pode-se observar isso, por exemplo, entre as curvas das sequências de testes 1 e 5, onde as faixas de variação destas duas curvas não se interceptam.

A mesma questão também foi observada quando a PUF possui osciladores com mais células de atraso (10 celulas). O resultado do teste é mostrado na Figura 4.10, na qual se pode ver a tendência de concentração destas frequências ao redor de uma frequência central de forma ainda mais acentuada.

Uma vez que os dados não foram todos capturados no mesmo dia, é possível que diferentes condições de temperatura, tensão de alimentação, condição de interconexões sejam diferentes, causando tal diferença. Porém, é considerado neste trabalho que a faixa de variação da frequência dos osciladores é suficientemente pequena para que um sistema de autenticação possa identificar a frequência do oscilador através de várias medições.

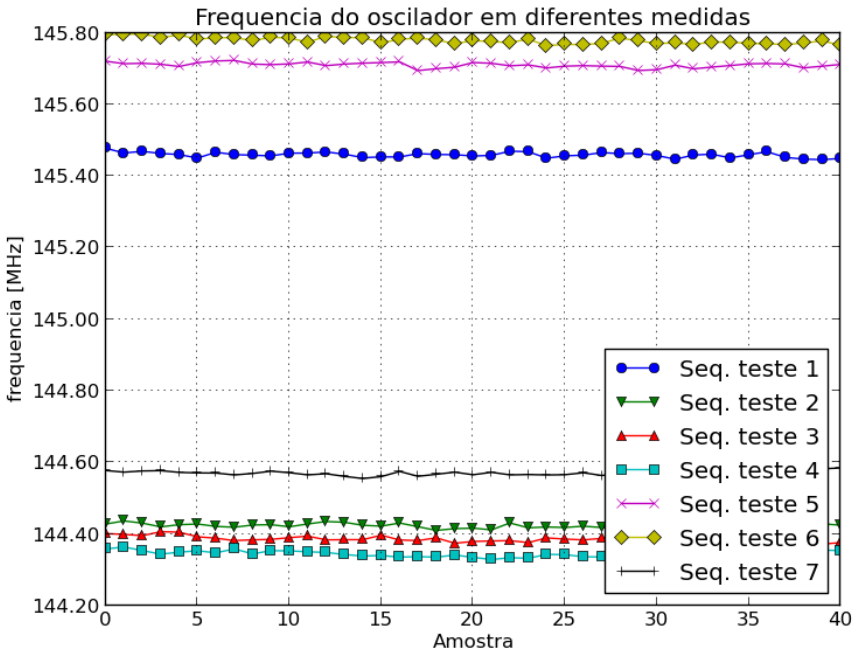

Figura 4.10 - Distribuição de frequência para um oscilador com 10 células de atraso em diferentes sequências de medição [46].

Após avaliar a variação da frequência de um oscilador em diferentes instantes de tempo, o próximo passo é avaliar o quão próximas são as frequências destes osciladores diferentes dentro de uma mesma PUF. Para isso, a frequência média de cada oscilador em cada sequência de medição necessita ser calculada, juntamente com o seu desvio padrão

percentual, o que foi feito e normalizado pela média das medições. Na Figura 4.11 é apresentada as frequências médias e os desvios padrão percentuais, normalizados pela frequência média de cada oscilador, para 256 osciladores em 6 sequências de medição diferentes. Todos os osciladores vistos têm 5 células de atraso.

Pode-se observar também que várias sequências de medição foram feitas, sendo cada uma delas uma curva do gráfico. Porém as médias das frequências dos osciladores não variam o suficiente para diferentes medições permitindo que as curvas possam ser discernidas visualmente. Por esta razão, foi feito uma ampliação em alguns pontos para que se possa verificar que de fato há diferentes curvas no gráfico, cada uma sendo uma sequência de medição diferente, embora sejam semelhantes. Foi feito também o mesmo teste, só que desta vez com osciladores com 10 células de atraso. Estes resultados são mostrados na Figura 4.12.

O primeiro ponto a ser observado nas Figuras 4.11 e 4.12 é que, embora os osciladores tenham sido concebidos com o mesmo projeto, há uma faixa de variação aproximada entre 220MHz e 300MHz, ou seja, de 36% com relação ao mínimo, para os osciladores com 5 células de atraso e uma faixa entre 140MHz a 155MHz, ou seja algo em torno de 10% com relação ao mínimo para os osciladores com 10 células de atraso. Isso é um indicativo de que o número de células de atraso dos osciladores influencia na entropia de saída da PUF, ou seja, na distribuição das frequências nos osciladores da mesma PUF. Quando se aumenta o número de células de atraso de 5 para 10, o desvio padrão percentual diminui. Este resultado é indicado nas Figuras 4.9 e 4.10 quando o aumento de células concentrava as medições de frequência em torno de uma frequência central, porém agora nas Figuras 4.11 e 4.12 tem-se um resultado quantitativo também. Outro fato interessante mostrado nos gráficos é que o desvio padrão não possui uma distribuição uniforme. Há certas regiões em que o desvio padrão assume valores menores do que em outros e isso indica que cada oscilador pode ser mais ou menos suscetível a variações, ou seja, é mais sensível a condições externas. Isso pode ser um fator complicador nos esquemas de compensação térmica. Em [52], o autor cita os chamados bit flips, que acontecem quando dado dois osciladores A e B, em que a frequência de A é maior do que a de B em determinada temperatura mas é menor do que B em outra faixa de temperatura. O ideal seria ter a relação entre frequência e temperatura para todos os osciladores individualmente, uma vez que suposições de que os osciladores variam de forma idêntica não são precisas.

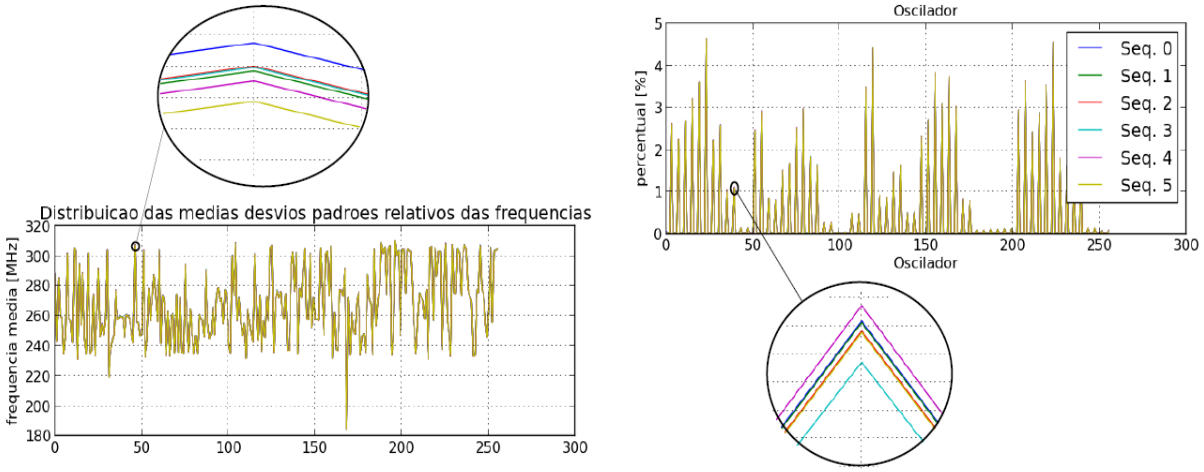

Figura 4.11 - Média (frequência) e desvio padrão percentual (percentual) em PUF de osciladores com 5 células de atraso [44].

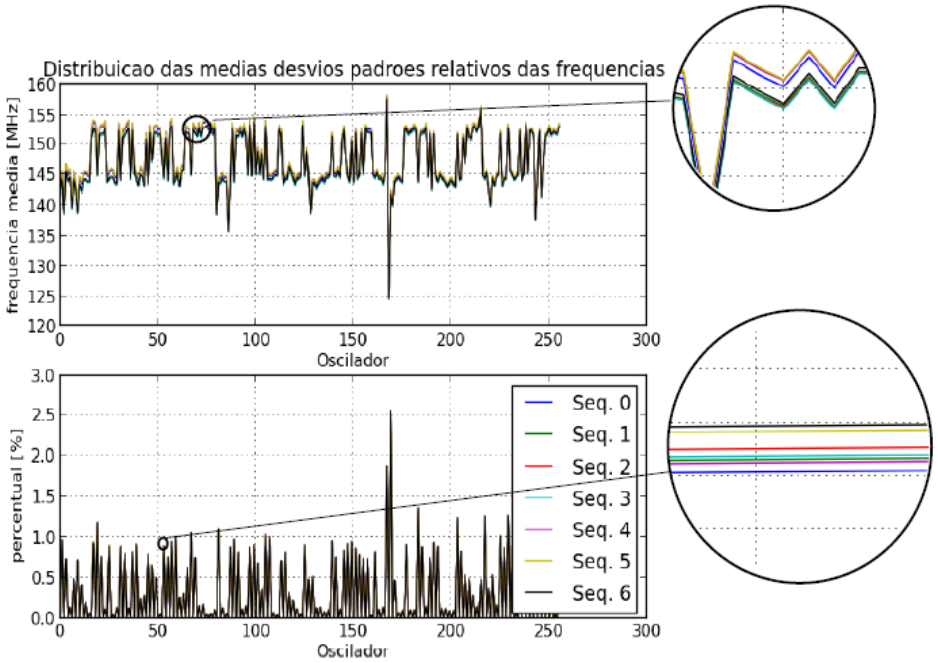

Figura 4.12 - Média (frequência) e desvio padrão percentual (percentual) em PUF de osciladores com 10 células de atraso [44].

Uma vez verificado que sequências de medição diferentes em instantes de tempo diferentes dos mesmos osciladores produzem assinaturas próximas entre si, a próxima pergunta a ser feita é o quão próximas duas assinaturas podem ser. Conforme visto no Capitulo 2 quando as caraterística mais importantes das PUFs foram abordadas, a proximidade das assinaturas está associada com a distância intra-classe da PUF, que avalia o quão parecido são as respostas para um o mesmo desafio para a mesma PUF. A métrica usada para se calcular este parâmetro é o erro absoluto normalizado entre estas assinaturas (respostas), ou seja, duas assinaturas obtidas com o mesmo desafio (oscilador escolhido) são

subtraídas uma da outra e o valor absoluto é calculado. Esse valor de diferença é normalizado pelo valor de uma das duas primeiras assinaturas. Na Figura 4.13, pode-se observar a assinatura de uma sequência de medição como referência para ser comparada com as outras. Essa assinatura também é utilizada na normalização dos valores.

Podem ser vistos na Figura 4.13 os valores correspondentes aos erros absolutos de algumas sequências de medições em comparação com uma referência. Devido à aglomeração de curvas na imagem, uma ampliação foi feita para auxiliar na visualização. Nessa imagem, os erros assumem, pelo menos visualmente, uma distribuição uniforme. Pode-se notar também que os erros não ultrapassam 0,2%.

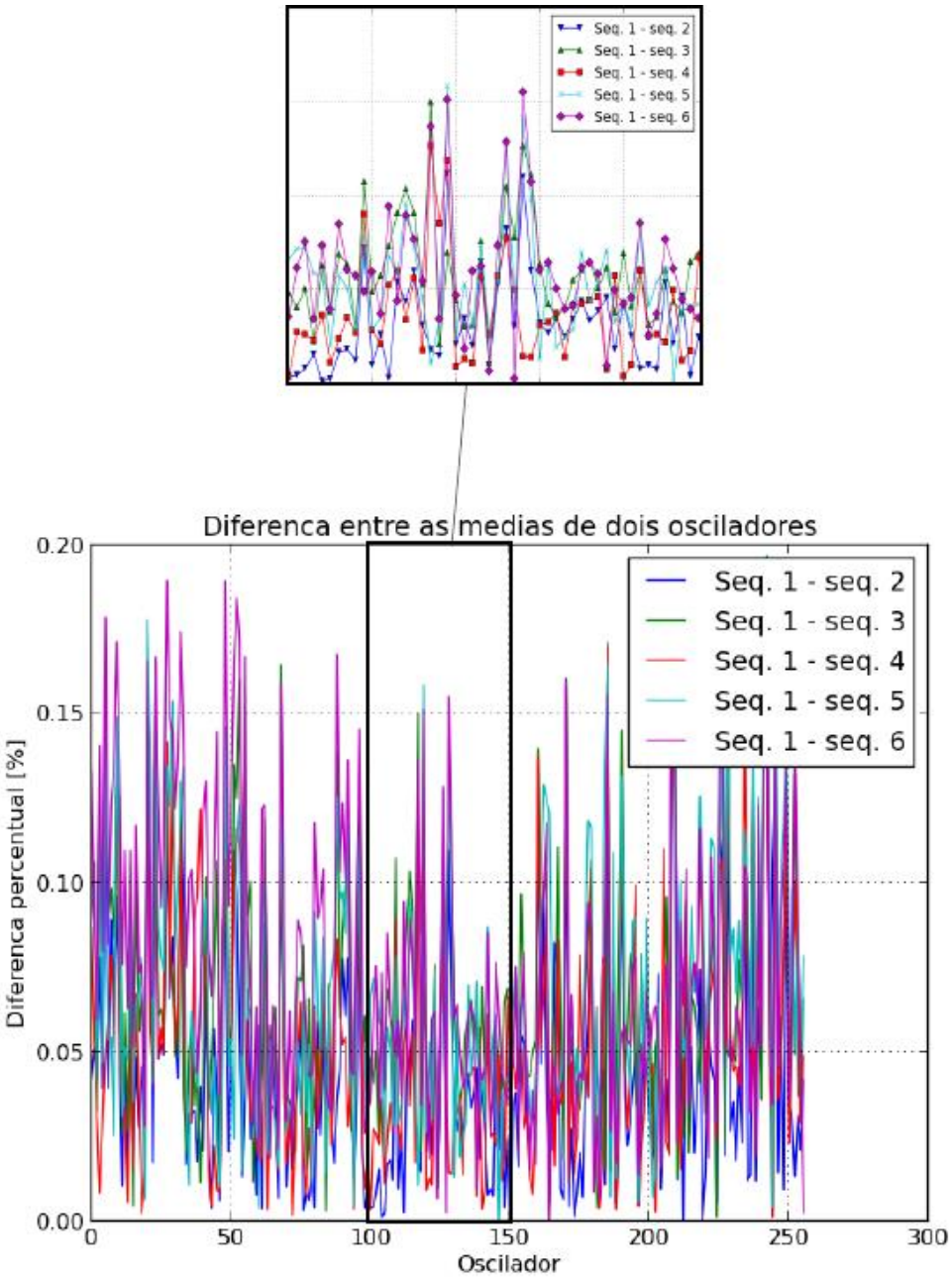

Figura 4.13- Diferença das assinaturas de duas sequências - PUF com 5 células de atraso

Os erros também foram calculados para o caso em que os osciladores da PUF são formados por 10 células de atraso. Pode-se notar com base na Figura 4.14 que quando o número de células de atraso aumenta para 10 células, há a formação de um padrão decrescente, mostrando que o erro em muitos casos é maior nos primeiros osciladores.

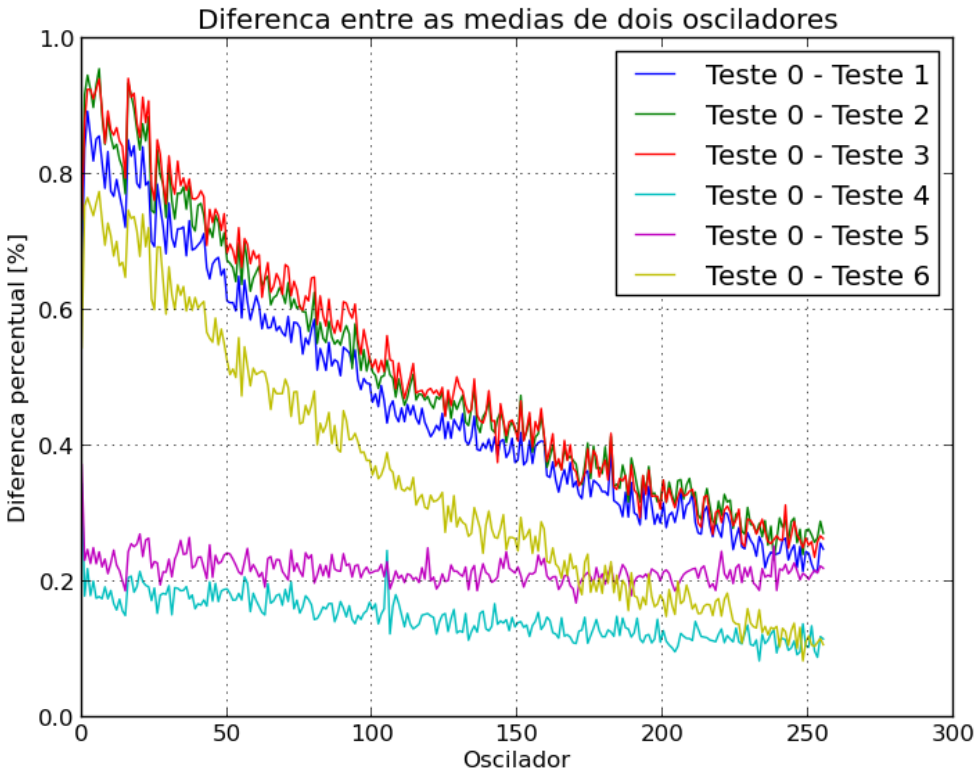

Figura 4.14- Diferença das assinaturas de duas sequências - PUF com 10 células de atraso

Mesmo com a formação de um padrão decrescente, observa-se que o erro obtido entre duas assinaturas é pequeno quando o mesmo arquivo de configuração é usado (intra-classe). Porém, quando diferentes arquivos de configuração são utilizados, é esperado que a PUF tenha seu comportamento alterado, uma vez que a cada alteração se utilizam diferentes componentes internos (inter-classe). Para utilizar-se elementos diferentes (células internas no FPGA), deve-se utilizar diferentes *bitstreams*, ou seja, diferentes arquivos de configuração (PUFs diferentes) do circuito oscilador em anel na FPGA.

Na Figura 4.15 são mostradas as diferentes assinaturas para cada PUF. Note que há 3 pares de curvas e em cada par há uma linha contínua e uma linha tracejada. Cada par de curvas mostra um par de assinaturas de PUFs feitas com o mesmo projeto, porém possuem *bitstreams* diferentes. No caso particular da Figura 4.15, como a configuração de quantidade de células é paramétrica, foi possível plotar também uma avalição com 20 células de atraso, apresentando que, quanto maior o numero de células mais plano é a variação de frequência

nos osciladores. É possível notar que, em comparação com as Figuras 4.11 e 4.12, as curvas contínuas e tracejadas não se sobrepõem tão claramente.

Isso é indicativo de que o erro será maior do que o mostrado nas Figuras 4.13 e 4.14. onde os erros não ultrapassam 1%.

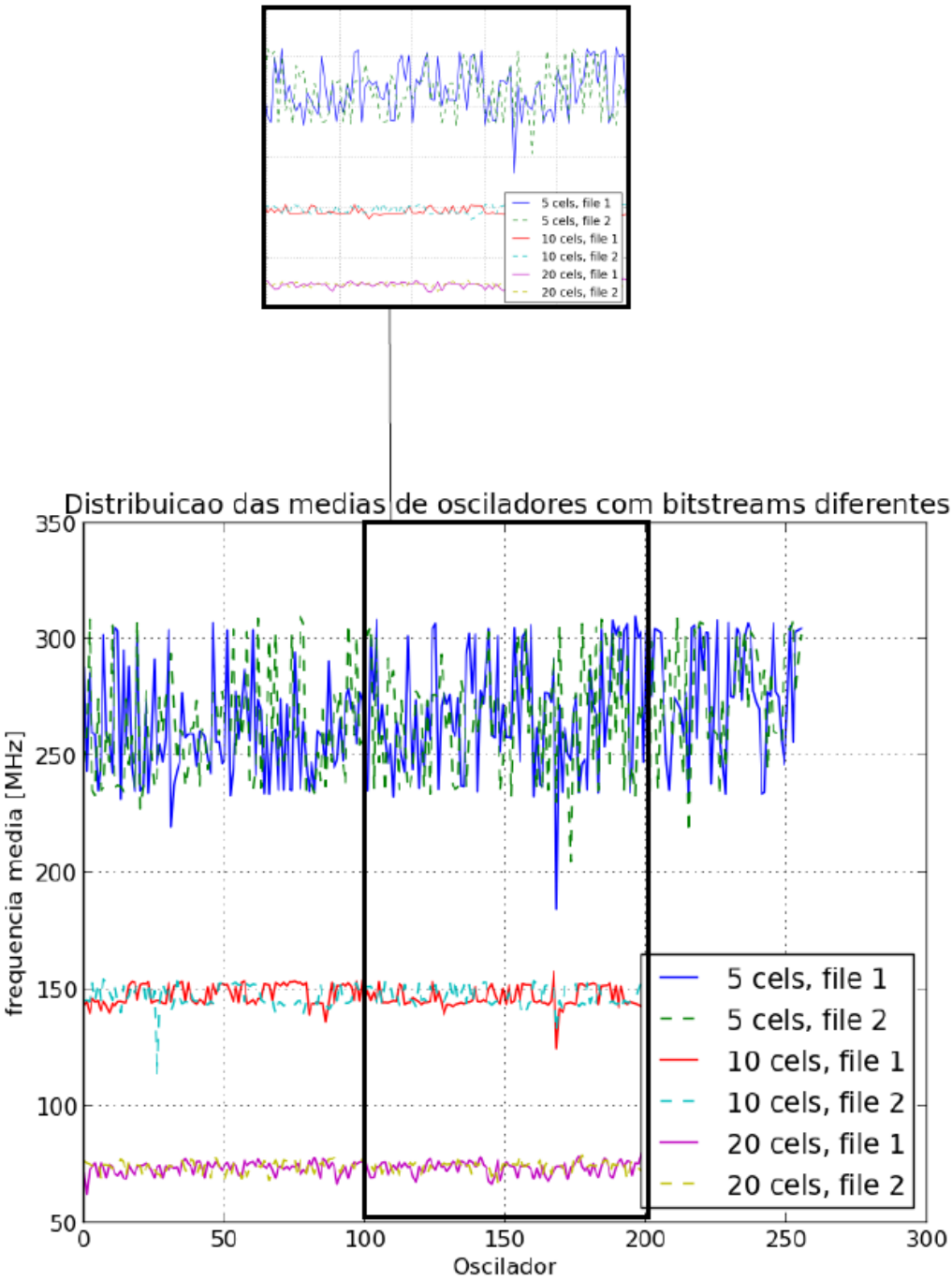

Figura 4.15 - Médias das frequências entre dois testes com *bitstreams* diferentes.

Quando em casos mais extremos são usados dois *bitstreams* diferentes, conforme se vê na Figura 4.16, os erros ultrapassam 20%, podendo atingir até quase 35%. Vale verificar que a

maioria deles fica acima de 5%. Isso indica que apesar de se utilizar a mesma FPGA, caso a região de logica programável não seja bem definida, os resultados irão variar consideravelmente, o que é desejável numa PUF, pois caso esta sofra uma tentativa de ataque por modelagem, será necessário que o atacante modele diferentes regiões da FPGA, tornando impossível a replicação da PUF.

Para que esse resultado possa ser confirmado quantitativamente, o erro absoluto das assinaturas deve ser computado da mesma maneira que foi computada quando se usou o mesmo *bitstream* de configuração. O resultado desse teste pode ser visto na Figura 4.16.

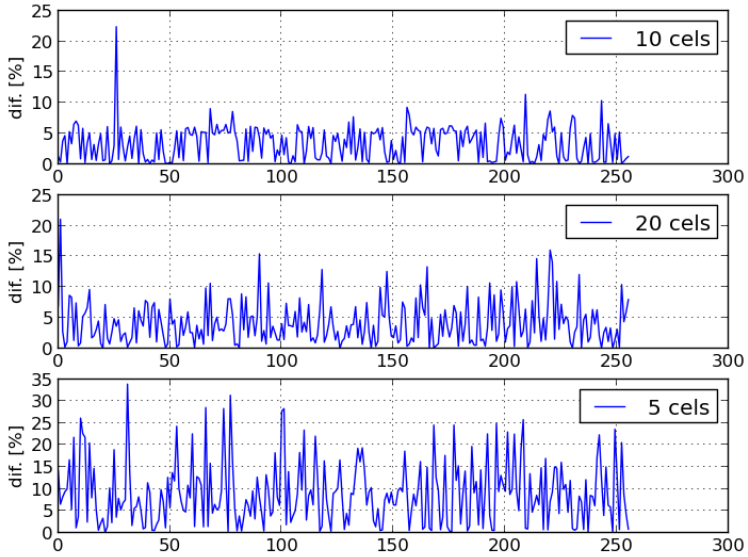

Figura 4.16 - Diferença das assinaturas de medição com *bitstreams* diferentes

Conforme visto, tem-se finalmente a extração da informação que é utilizada como assinatura final, ou seja, é a informação definitiva que será armazenada no vetor numérico e será utilizada como assinatura para a geração de chave de autenticação do componente semicondutor. Nesta fase se finaliza a proposta da geração de um número aleatório com a maior probabilidade possível de não repetitibilidade, atingindo assim o objetivo de uso de PUFs para identificação única.

4.3 Blockchain Utilizada

Neste trabalho, o foco é a introdução alguma solução nova e possivelmente disruptiva, que possa ter o potencial de transformar a segurança no acesso a dispositivos com base em semicondutores. Dito isto, e com a intenção de atuar especificamente em uma solução com foco em autenticar um dispositivo IoT e validar se as informações enviadas são confiáveis, esta trabalho incrementa o processo de segurança e autenticação local baseada em PUFs e trás

para o cenário de segurança de dados a figura da confiança e dupla autenticação, uma local e outra distribuída.

Com foco na nova tecnologia de blockchain apresentada anteriormente, apresenta-se aqui uma proposta de solução de autenticação e como esta deve ser implementada. Assim surge a necessidade de se discutir a respeito de *smartcontracts* (contratos inteligentes).

Apesar da *blockchain* permitir transacionar ativos sem autenticação de confiança de uma terceira parte, é necessário também um processo que registre um ativo e o valide nesta rede. Mais do que isto, seria interessante também que ações futuras como a atualização de software embarcado, posicionamento georeferenciado do dispositivo, entre outros, pudessem ser registrados além da implementação de um processo de execução de ações baseadas em cenário de acordo prévio.

Com isto, tem-se agora a necessidade, não apenas de se implementar uma *blockchain* qualquer que apenas autentica transações, mas sim, uma que implemente um modelo de governança baseado em contratos inteligentes. Assim, conforme discutido e demonstrado no Capitulo 3, a solução mais viável e segura é rede *blockchain* Ethereum.

4.3.1 Modelo operacional (envio de informação e tratamento)

Neste trabalho temos uma solução de contratos inteligentes abrigando as demandas e necessidades da aplicação inicial, ou seja, temos as funções para registrar um novo dispositivo (ASIC/FPGA) e autentica-lo remotamente. Aqui, estão descritos todos os elementos necessários para a implementação do contrato inteligente e sua conexão com a identificação única do CI (PUF). Como visto em [53] primeiramente se faz necessário criar as chaves e endereços de propriedade para identificar exclusivamente um componente como um novo ASIC ou mesmo uma FPGA já produzida. No caso deste trabalho temos uma solução de lógica programável, conforme descrito no capítulo 2, para se evitar qualquer falsificação e/ou problemas de autenticação do mesmo. A todos os componentes são atribuídos pares de chaves públicas e privadas.

O proprietário pode gerenciar chaves e endereços pessoais usando uma carteira digital. A carteira digital pode ser usada para realizar qualquer transação. Uma carteira digital é um software ou hardware projetado especificamente, que contém as chaves públicas-privadas e o endereço do componente (Id). O contrato inteligente, cujo fluxo pode ser visto na Figura 4.17, será criado pelo proprietário da solução de segurança ou fabricante, para manter aplicabilidade e usabilidade uniformes para todos os dispositivos. Neste caso, o contrato

inteligente fornece os serviços de registro de dispositivo, verificação das informações e mudança de propriedade e autenticação.

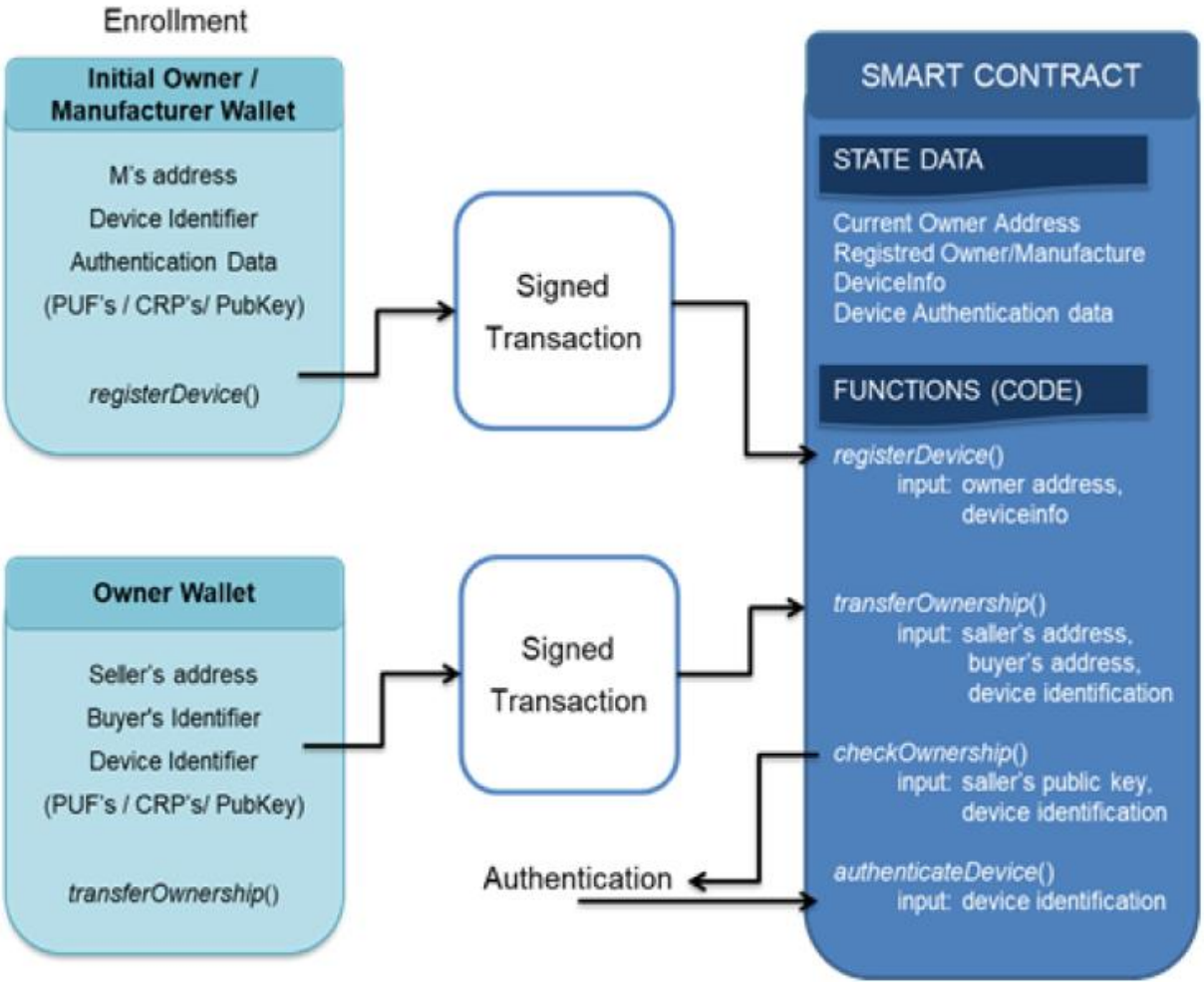

Figura 4.17 - Fluxo de Implementação do Contrato Inteligente

Registro do dispositivo - Para introduzir um dispositivo na *blockchain*, ele deve ser primeiro registrado [54]. Como indicado na Figura 4.18, o pseudocódigo da função que implementa o registro de contrato inteligente é a *registerDevice*. Esta função registra um dispositivo se o remetente da mensagem for o proprietário da solução de segurança ou fabricante. A chamada *deviceInfo*, inclui dados para identificação e autenticação do componente. Os dados de identificação são utilizados para pesquisar, entre uma coleção de dispositivos, o *device* de destino que está sendo consultado. Pode ser, por exemplo, um número de série do dispositivo, o código de produto eletrônico ou outro identificador específico. Este dado do dispositivo é usado para fins de identificação e não como o meio principal de autenticação. Para registro inicial, o fabricante usa as informações do componente (*deviceInfo*) como entrada para a função *registerDevice*.

```
inputs: manufacturer's address (addrManufacturer),
        and device information (deviceInfo)
if message sender is in manufacturer's list them
|       specify owner of the device as addrManufacturer
|       register deviceInfo on de blockchain
else
|       do nothing
end
```

Figura 4.18 - Pseudo Código para o registro (função *registerDevice*).

Verificação de Propriedade - A propriedade do dispositivo pode ser verificada pela função do contrato inteligente *checkOwnership* indicado na Figura 4.19, onde temos seu pseudocódigo. Esta função verifica a propriedade do dispositivo em relação ao endereço do proprietário. Se o dispositivo com o identificador fornecido pertencer ao remetente, ele retornará *True*. A função é chamada quando o proprietário do dispositivo desejar confirmar a origem de registro.

```
inputs: seller's public key (sellerPubKey), and
        device identifier (deviceIdentifier)
outputs: a boolean True or False
if (hash(sellerPubKey) ==
   blockchain[deviceIdentifier].owner) them
|       return True
else
|       return False
end
```

Figura 4.19 - Pseudo Código para verificação de propriedade (função *checkOwnership*)

Transferência de Propriedade - A transferência segura de propriedade se faz necessária caso o cliente final deseje implantar por si próprio sua solução de autenticação ou aquisição de dados. Para se transferir a propriedade de um dispositivo, o contrato inteligente implementa a função *transferOwnership*, como indicado pelo pseudocódigo na Figura 4.20. Esta função transfere a propriedade do dispositivo (*deviceIdentifier*) do vendedor (*addrSeller*) ao comprador (*addrBuyer*). Primeiro, a função verifica se o remetente da mensagem é o proprietário do dispositivo com *deviceIdentifier*. Se isso for verdadeiro, a função atribui o resultado da função *addrBuyer* como o novo proprietário do dispositivo.

```
inputs: buyer's address (addrBuyer), and device
   Identifier (deviceIdentifier)
if (addrMessageSender ==
   blockchain[deviceIdentifier].onwer) them
 |    set blockchain[deviceIdentifier]. Owner = addrBuyer
else
 |    do nothing
end
```

Figura 4.20 - Pseudo Código para transferência de propriedade (função *transferOwnership*)

Autenticação do dispositivo - Qualquer sistema tentando se comunicar com um dispositivo deve confirmar que o dispositivo é autêntico. Isso é implementado pela função de contrato inteligente *authenticateDevice* como indicado pelo pseudocódigo da Figura 4.21. Esta função inicia o processo de autenticação do dispositivo para um determinado identificador. O processo basicamente executa funções para se obter os dados de um *deviceIdentifier* específico após aplicar um desafio ao dispositivo, calcular a resposta e combinar os pares desafio-resposta, verificando assim a autenticidade ou não do mesmo.

```
inputs: identifier of the device (deviceIdentifier)
outputs: a boolean True or False
set deviceChallenge = blockchain[deviceIdentifier].Challenge
get deviceResponse from the device after applying
   deviceChallenge
if (deviceResponse ==
   blockchain[deviceIdentifier].Response) them
 |        return True
else
 |        return False
end
```

Figura 4.21 - Pseudo Código para Autenticação (função *authenticateDevice*).

4.4 Protocolo de Autenticação e Transmissão Segura de Dados

Descreve-se a seguir, conforme mostrado na Figura 4.22, o protocolo objeto desta tese, integralizado tanto no *device* quanto a blockchain, para a autenticação de um dispositivo e a transmissão segura de suas informações. Pode-se considerar, por exemplo, a transmissão de dados entre uma solução de sensoriamento ambiental (IoT *device*) e uma plataforma de software para aquisição de dados (*dashboard*).

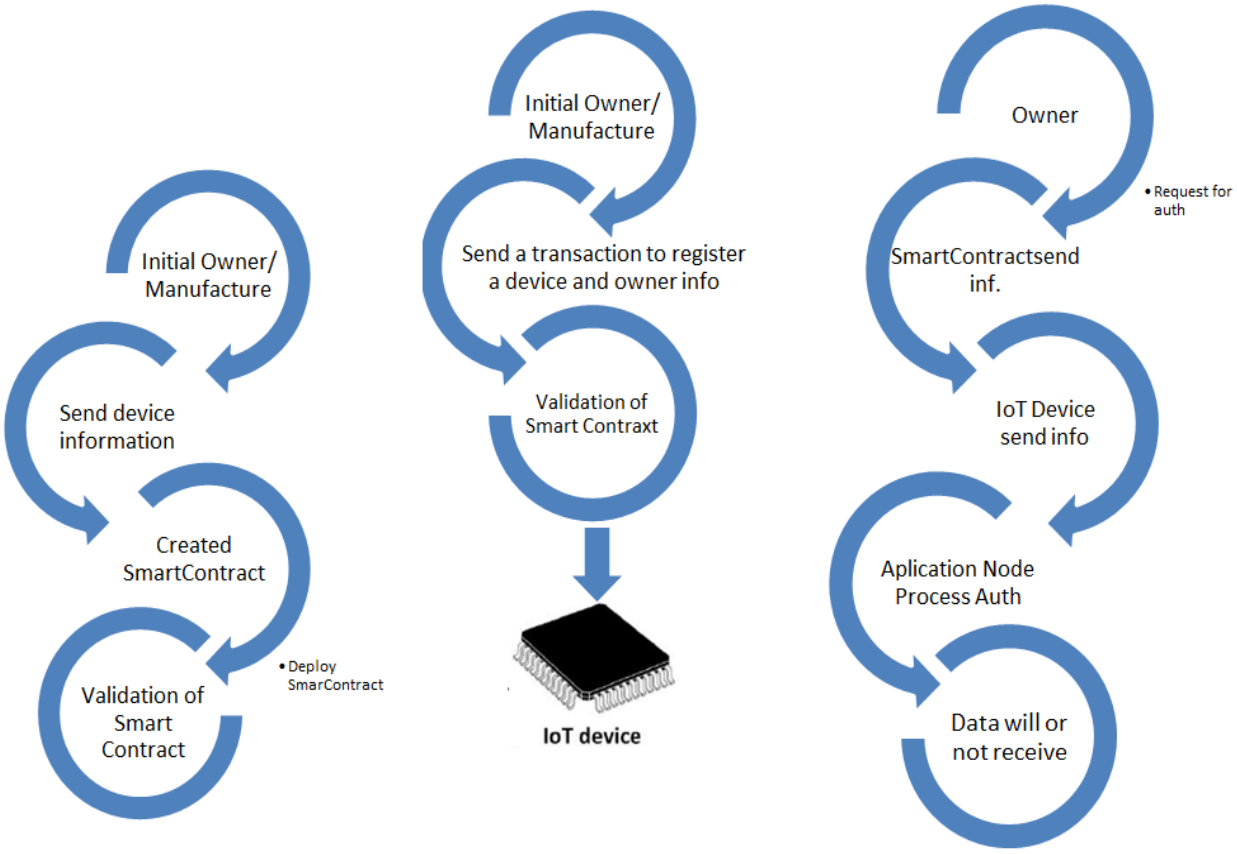

Figura 4.22 - Diagramas para registro, validação, autenticação e transmissão de dados

Criação de contrato inteligente para configuração de recursos e atualização de firmware - Para se automatizar o processo de autenticação do dispositivo, o fabricante primeiro cria um contrato inteligente. O proprietário da solução de segurança ou fabricante envia as informações de iniciais do dispositivo para nó blockchain com a finalidade de se criar o contrato. As informações do dispositivo incluem a *wallet* do proprietário, Id do dispositivo e dados de autenticação (*Public Keys*, etc.).

Implantação de contrato inteligente para configuração de recursos de autenticação - O nó do proprietário da solução de segurança ou fabricante, cria na blockchain o contrato inteligente. Depois de criar o contrato inteligente o nó do fornecedor o implanta na *blockchain*. Depois de validado pelos nós, o contrato inteligente é adicionado a *blockchain* como uma transação e um endereço é atribuído a ele. Após essa transação inicial, o contrato torna-se parte da *blockchain* para sempre e seu endereço nunca muda. O dono do registro incorpora este endereço no módulo de segurança do dispositivo como indicado na Figura 4.23.

Inscrição de um dispositivo pelo proprietário ou fabricante - Quando um proprietário ou fabricante deseja registrar um dispositivo, ele envia uma transação *registerDevice* para o *smartcontract* (Figura 4.17). O fabricante também envia as informações do dispositivo, como identificador do dispositivo, informações de autenticação da transação, entre outros. Para efeito deste trabalho, denomina-se a primeira transação de registro de um dispositivo como transação "gênesis". O dono também pode registrar um número N de dispositivos na mesma transação, incluindo as informações correspondentes de todos os dispositivos. Isso facilita a escalabilidade do protocolo. Como todas as transações da *blockchain* são assinadas digitalmente pelo ator que as cria, um possível falsificador não consegue alegar de forma fraudulenta ser um fabricante e alterar o registro.

Solicitação de informações autenticadas de dispositivo - Usando o endereço embutido no módulo de segurança, o proprietário de um dispositivo envia uma solicitação de autenticação pelo seu sistema proprietário (plataforma de software privada), para o contrato inteligente criado na *blockchain*, conforme indicado na Figura 4.23. O nó *blockchain* recebe a solicitação e verifica se os requisitos no contrato correspondem à solicitação recebida. Se os requisitos forem satisfeitos, o nó envia a sinalização correspondente para o sistema proprietário para autenticação do dispositivo solicitante. Antes de enviar a autenticação, o contrato inteligente verifica a propriedade do dispositivo solicitante.

4.4.1 Modulo de segurança em silício

Na figura 4.23 temos o diagrama que implementa a solução do modelo de autenticação em CI apresentado neste trabalho. Veremos então as funcionalidades e o funcionamento dos principais blocos funcionais.

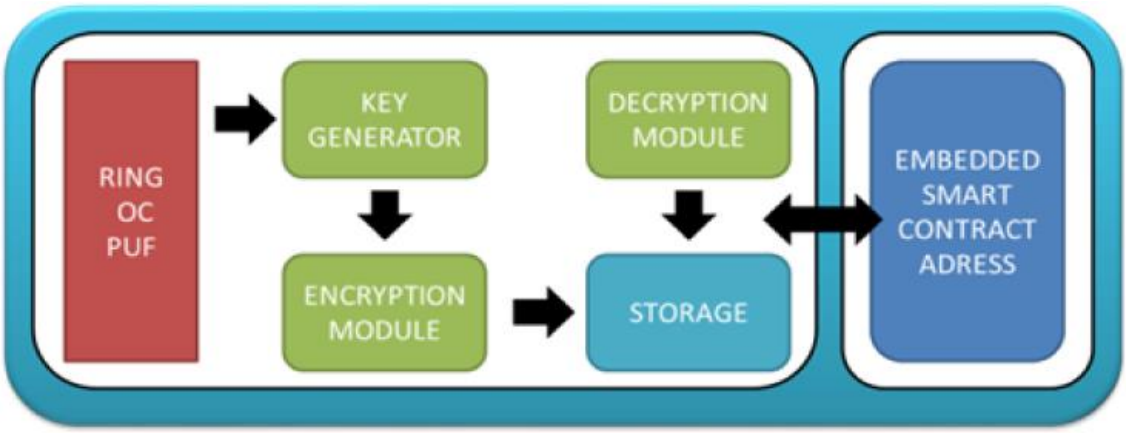

Figura 4.23 - Diagrama Detalhado do Modulo de Segurança.

Verificando as informações de propriedade - Para se verificar as informações de propriedade atuais, o sistema de validação de dispositivo invoca a função *checkOwnership* do contrato inteligente. Se o retorno for *True*, a propriedade do dispositivo é verificada.

Autenticação do IC - Verificar a propriedade de um dispositivo não é suficiente. Um invasor mal-intencionado pode substituir o dispositivo original por um falsificado e solicitar atualização de recursos para o falsificado. Portanto, o dispositivo também precisa ser autenticado com as informações armazenadas na *blockchain*. Neste momento utiliza-se a chave pública baseada em PUF para autenticação. A Figura 4.23 apresenta um módulo de autenticação de hardware para gerar uma chave criptográfica a partir de uma PUF baseada em osciladores em anel. O fabricante primeiro registra os dados da PUF na *blockchain* emitindo uma transação *registerDevice* em qualquer estágio da cadeia de suprimentos, o dispositivo pode ser autenticado invocando *authenticateDevice*.

Os componentes do módulo de autenticação e suas funções são descritos a seguir:

a) Módulo de geração de chave - Durante a fase de inscrição na plataforma de sistema do proprietário ou fabricante, o gerador de chaves no módulo de autenticação gera um par de chaves públicas e privadas. Usando a chave privada, um dispositivo pode criar a assinatura de uma mensagem protegendo a integridade da mensagem e comprovando sua autenticidade. No final do recebimento da mensagem, a autenticidade da assinatura digital pode ser verificada usando a chave pública correspondente à chave privada. O par de chave pública-privada podem ser chaves RSA, DSA, sistemas criptográficos baseados em Curva Elíptica, ou outros, basta se implementar, em nível de sistema, a solução desejada.

b) Módulo de criptografia - A chave privada é criptografada com uma assinatura única gerada a partir da PUF e armazenada em uma memória não volátil. Essa criptografia é feita usando, por exemplo, um algoritmo de criptografia simétrica (AES 256). O dono registra a chave pública no blockchain emitindo uma transação *registerDevice* para o dispositivo como indicado na Figura 4.24a.

c) Módulo de decriptografia - Durante a fase de autenticação, a chave de criptografia pode ser gerada instantaneamente a partir da saída da PUF. Usando a assinatura única da PUF, o módulo de decriptografia gera a chave privada do dispositivo.

O dispositivo pode ser autenticado invocando a função *authenticateDevice* com o seu identificador. O contrato inteligente envia uma mensagem de desafio, produzida com um gerador de números pseudoaleatórios cifrada com a chave publica do *device* para o mesmo. O módulo decriptográfico do dispositivo, usando a chave privada gerada a partir da assinatura

digital da PUF no modulo de geração de chave, decifra o numero aleatório gerado no *smartcontract*. O *device* devolve para o *smartcontract* uma mensagem assinada com o valor do numero aleatório gerado inicialmente, que após confirmação na *blockchain*, valida a autenticidade do dispositivo, conforme mostrado na Figura 25b.

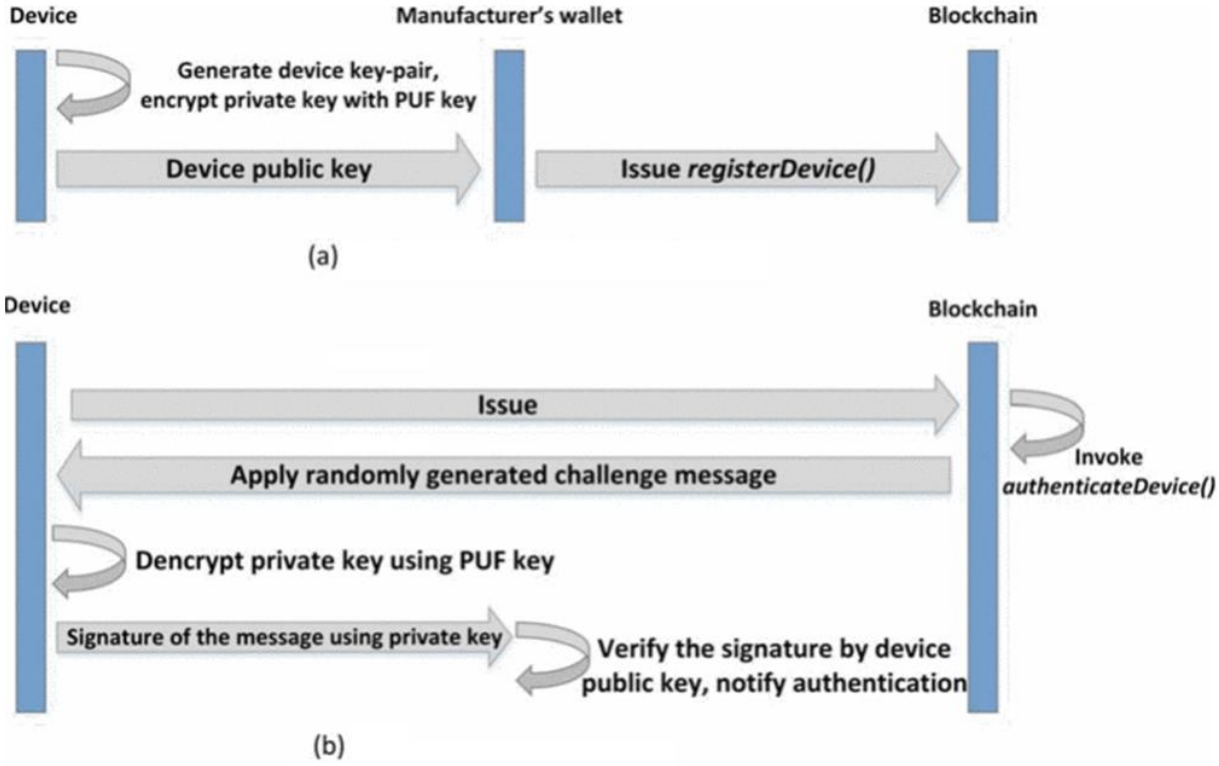

Figura 4.24 - Processo de cadastramento do dispositivo(a); Processo de autenticação(b).

Depois que o sistema proprietário (plataforma de software privada) faz uma solicitação de autenticação do *device* e assim a confirma, o componente deve preparar as informações para serem enviadas. Neste caso em particular é necessário que a informação também possa trafegar de forma segura pela rede.

Uma infraestrutura de comunicação deve implementar as propriedades da tríade CIA, ou seja, confidencialidade, integridade e disponibilidade. Assim, para atender ao quesito integridade da informação e autenticação do remetente, o sistema descrito aqui, cria uma chave criptográfica baseada na unicidade da assinatura digital da PUF do componente. Para manter a confidencialidade, o componente criptografa os dados a serem enviados com uma chave publica produzida no sistema proprietário (software de aquisição) e assim as transmite.

Em seguida, o sistema que solicitou a informação a descriptografa e os dados transmitidos podem ser utilizados de forma confiável, após a autenticação prévia conferida.

5 Conclusões

Conforme visto, este trabalho apresenta um modelo funcional completo para a implantação de sistemas de autenticação de componentes semicondutores, sensores e outros. Este modelo pode ser utilizado não apenas para um circuito integrado único, mas também para um dispositivo eletrônico completo.

Descrevemos aqui desde a criação de uma função física não clonável (PUF), a extração de suas informações únicas, passando pelo modelo funcional do circuito integrado que processa e produz as garantias de autenticidade, requisições e transmissão de dados, até a aplicação de alto nível que faz uso efetivo da proposta.

A inovação neste trabalho não esta relacionada na proposição de um PUF diferenciada que pode incrementar paradigmas de segurança ou ser resistente a novas formas de tentativas de *hacking*, nem na estruturação de uma plataforma intermediária (hardware + firmware), que implementa procedimentos de verificação e validação do componente, ou no sistema de alto nível. A contribuição e ineditismo deste trabalho se dão na integração de tecnologias recentes e maduras, e que exploram características singulares nos campos do software e da microeletrônica, de modo alcançar uma nova forma de autenticação e segurança cibernética para semicondutores e dispositivos de forma mais simples e descentralizada.

No final, a intenção não é apresentar uma implementação completa de uma solução com toda a sua plataforma funcional ou um sistema com simulação e validação de dados para comparações com outras soluções, mas sim colocar um modelo muito qualificado e detalhado para estimular a sua implementação final e trabalhos futuros. Por isto este trabalho não realizou uma avaliação detalhada de custos econômicos, complexidade computacional ou desempenho. Será necessário implementar um modelo funcional completo, testes e simulações.

Além disso, a solução de software apresentada neste trabalho não está implementada em uma linguagem de programação específica como Solidity para a *blockchain* Ethereum, pois é necessário considerar uma visão mais ampla em pseudo-código que pode ser desenvolvida em outras plataforma de contratos inteligentes em *blockchains* mais recentes, como Cardano, Polkadot, Binance Smart Chain, Fantom, etc.

Por fim, entendemos que o valor está na proposta do modelo com a integração das soluções, apontando caminhos específicos, ganhos em simplicidade, facilidade de implementação e por consequência custos implicados, demonstrando desta forma sua viabilidade, promovendo assim a inovação tão desejada.

Apêndice A – Trabalhos Publicados.

CAMPOS, A. A. N; PIMENTA, T. C. Authentication for Integrated Circuit and Devices using Blockchain and Physical Unclonable Functions. Journal of Integrated Circuits and Systems (JICS), May 2022. ISSN: 1872-0234. 11 pages. DOI:https://doi.org/10.29292/jics.v17i1.555

Referências

[1] Layton, T. P. “Information Security: Design, Implementation, Measurement, and Compliance”, CRC Press, 2016 ISBN: 1420013416, 9781420013412; 2016.

[2] Editors: Benedikt Gierlichs, Axel Y. Poschmann. “Cryptographic Hardware and Embedded Systems – CHES 2016” 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings, Springer, 2016 ISBN: 3662531402, 9783662531402; 2016.

[3] Kelly, S.; Zhang, X.; Tehranipoor, M.; Ferraiuolo, A. “Detecting Hardware Trojans using On-chip Sensors in an ASIC Design” Springer - Journal of Electronic Testing - February 2015, Volume 31, Issue 1, pp 11–26; 2015.

[4] Grauman, B. “Cyber-security: The vexed question of global rules” Bruxelas. SDA – Security & Defense Agenda , 2013

[5] BRASIL. Livro Branco de Defesa Nacional. Brasília Presidência da República, 2012.

[6] Da Cruz Jr., S.C. “A segurança e a defesa cibernética no Brasil e uma revisão das estratégias dos Estados Unidos, Rússia e India para o espaço virtual” Texto para Discussão 1850. Brasília: IPEA, 2013.

[7] Saxena, M.; Kumar, R. “A recent trends in software defined networking (SDN) security”, 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016.

[8] M. Bishop “What is computer security?“ EEE Security & Privacy, Issue 1 • Jan.-Feb. 2003

[9] Bossuet, L.; Colombier, B. “A PUF-FSM Binding Scheme for FPGA IP Protection and Pay-per-Device Licensing” IEEE Transactions on Information Forensics and Security; Issue 11 • Nov. 2016.

[10] Parrillaa, L.; Castilloa, E.; Todorovichb, E.; Garcíaa, A.; Moralesa, D.P.; Botellac. G. “Improvements for the applicability of power-watermarking to embedded IP cores protection: e-coreIPP” Elsevir Digital Signal Processing Volume 44, September 2015, Pages 110–122; 2015.

[11] Shahed E. Quadir, Junlin Chen, Domenic Forte, Navid Asadizanjani, Sina Shahbazmohamadi, Lei Wang, John Chandy, and Mark Tehranipoor. “A survey on chip to system reverse engineering”. ACM Journal on Emerging Technologies in Computing Systems,. 13, 1, Article 6 (April 2016), 34 pages. DOI: http://dx.doi.org/10.1145/2755563

[12] D. F. M. M. T. Q. S. Huanyu Wang “Probing Attacks on Integrated Circuits: Challenges and Research Opportunities” IEEE Design & Test , October 2017.

[13] Helfmeier C., Boit C., Nedospasov D., Seifert J. P. “Cloning Physically Unclonable Functions” In IEEE International Symposium on Hardware-Oriented Security and Trust (HOST 2013)., pages 1-6. IEEE, June 2013.

[14] Pappu, R.S. “Physical one-way functions” Ph.D. thesis, Massachusetts Institute of Technology (2001).

[15] Pappu, R.S., Recht, B., Taylor, J., Gershenfeld, N. “Physical one-way functions” Science 297, 2026/2030 (2002).

[16] Bauder, D. “An anti-counterfeiting concept for currency systems” Tech. Rep. PTK-11990, Sandia National Labs, Albuquerque, NM (1983).

[17] Commission on Engineering and Technical Systems (CETS): Counterfeit Deterrent Features for the Next-Generation Currency Design. The National Academic Press (1993). Appendix E.

[18] Bulens, P., Standaert, F.X., Quisquater, J.J. “How to strongly lonk data and its medium: the paper case” em: IET Information Security (to appear) (2010).

[19] Vrijaldenhoven, S.; “Acoustical Physical Uncloneable Functions” Master's thesis, Technische Universiteit Eindhoven, the Netherlands (2005).

[20] Lofstrom, K., Daasch, W.R., Taylor, D. “IC Identification Circuit Using Device Mismatch” em: In Proceedings of ISSCC 2000, pp. 372{373 (2000).

[21] Tuyls, P., Schrijen, G.J., Skoric, B., van Geloven, J., Verhaegh, N., Wolters, R. “Readproof hardware from protective coatings” em: Cryptographic Hardware and Embedded Systems Workshop, LNCS, vol. 4249, pp. 369{383. Springer (2006).

[22] Lee, J.W., Lim, D., Gassend, B., Suh, G.E., van Dijk, M., Devadas, S. “A technique to build a secret key in integrated circuits for identication and authentication application” em: Proceedings of the Symposium on VLSI Circuits, pp. 176{159 (2004).

[23] Lim, D. “Extracting Secret Keys from Integrated Circuits” Master's thesis, MIT, MA, USA (2004).

[24] Gassend, B. “Physical Random Functions” Master's thesis, MIT, MA, USA (2003).

[25] Gassend, B., Clarke, D., van Dijk, M., Devadas, S. “Silicon physical random functions” Em: ACM Conference on Computer and Communications Security, pp. 148{160. ACM Press, New York, NY, USA (2002).

[26] Guajardo, J., Kumar, S.S., Schrijen, G.J., Tuyls, P. “FPGA intrinsic PUFs and their use for IP protection” em: Cryptographic Hardware and Embedded Systems Workshop, LNCS, vol. 4727, pp. 63{80 (2007).

[27] Holcomb, D.E., Burleson, W.P., Fu, K.”Initial SRAM state as a fingerprint and source of true random numbers for RFID tags” em: Proceedings of the Conference on RFID Security (2007).

[28] Y. Su, J. Holleman and B. P. Otis, "A Digital 1.6 pJ/bit Chip Identification Circuit Using Process Variations," in IEEE Journal of Solid-State Circuits, vol. 43, no. 1, pp. 69-77, Jan. 2008, doi: 10.1109/JSSC.2007.910961.

[29] Maes, R., Tuyls, P., Verbauwhede, I. “Intrinsic pufs from flip-flops on recongurable devices” em: 3rd Benelux Workshop on Information and System Security (WISSec 2008). Eindhoven,NL (2008).

[30] Pullanikkat, S. & Perumal, M. ”Design of silicon Physical Unclonable Function for authentication of a multicore device” em: Research Journal of Pharmaceutical, Biological and Chemical Sciences. 7. 933-941.

[31] Nakamoto, S. “Bitcoin: A Peer-to-Peer Electronic Cash System” em: Bitcoin Org, 2008. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em: 20 out. 2020

[32] “Hyperledger Blockchain Performance Metrics” em: Hyperledger Org 2018. Disponível em: https://www.hyperledger.org/learn/publications/blockchain-performance-metrics#. Acesso em: 20 out. 2020.

[33] Buterin V. “A next-generation smart contract and decentralized application platform” em: Ethereum Org, 2014. Disponível em: https://ethereum.org/en/whitepaper/ Acesso em: 20 out. 2020.

[34] PIRES, H. F. “Bitcoin: a moeda do ciberespaço” GEOUSP Espaço e Tempo, [S. l.], v. 21, n. 2, p. 407-424, 2017. DOI: 10.11606/issn.2179-0892.geousp.2017.134538. Disponível em: https://www.revistas.usp.br/geousp/article/view/134538. Acesso em: 8 maio. 2022.

[35] “Definição, funcionamento e as aplicações do hash, a função popular da criptografia”. em: Voitto Educação, 2020. Disponível em: https://www.voitto.com.br/blog/artigo/o-que-e-hash-e-como-funciona Acesso em: 30 nov. 2020

[36] Ferreira F. Lage “Blockchain e Ethereum - Aplicações e Vulnerabilidades” Universidade de São Paulo – USP - Instituto de Matemática e Estatística Bacharelado em Ciência da Computação – Trabalho de Conclusão de curso, 2017.

[37] Jakobsson, M., Juels, A. “Proofs of Work and Bread Pudding Protocols (Extended Abstract)”. em: Preneel, B. (eds) Secure Information Networks. IFIP The International Federation for Information Processing, vol 23. Springer, Boston, MA..

[38] “Aplicações da Tecnologia Blockchain” em: Dsacademy 2018 Disponível em: https://blog.dsacademy.com.br/aplicacoes-da-tecnologia-blockchain/. Acesso em: 08 maio. 2022.

[39] “O que são Altcoins?” em: EXAME 2020. Disponível em: https://exame.com/dinheiro-tendencias/o-que-sao-altcoins/. Acesso em: 08 maio. 2022.

[40] “Por que as Altcoins são muito melhores do que o Bitcoin??” em: LS Educação 2020. Disponível em: https://ls.com.vc/educacao/artigo/por-que-as-altcoins-sao-muito-melhores-do-que-o-bitcoin. Acesso em: 20 out. 2020.

[41] “O que é Ethereum (ETH)?” em: Stormgain 2020. Disponível em: https://stormgain.com/pt-br/blog/what-ethereum-eth. Acesso em: 24 out. 2020

[42] “Conheça cinco inovações que destacaram o Ethereum em 5 anos de história” em: Criptofacil 2020. Disponível em: https://www.criptofacil.com/conheca-cinco-inovacoes-que-destacaram-ethereum-5-anos-historia/. Acesso em: 24 out. 2020.

[43] “Solidity: a linguagem de programação para criar os smart contracts na Ethereum” em: Voitto 2020. Disponível em: https://www.voitto.com.br/blog/artigo/linguagem-de-programacao-solidity. Acesso em: 30 nov. 2020.

[44] “Criptografia – Paradigmas e Técnicas de Autenticação” em: Boson Treinamentos 2015. Disponível em: http://www.bosontreinamentos.com.br/seguranca/criptografia-paradigmas-e-tecnicas-de-autenticacao/. Acesso em: 08 maio. 2022

[45] “Biometria, Sistema Biométrico: O que é, Como Funciona?” em: Blog Gestao de Seguranca Privada 2015 Disponível em: https://gestaodesegurancaprivada.com.br/wp-content/uploads/biometria.jpg. Acesso em: 17 maio 2022.

[46] TREVISAN, G. V. “Análise de Physical Unclonable Functions baseadas em osciladores em anel em FPGA”. iv, 50 f., il. Monografia (Bacharelado em Engenharia Elétrica)—Universidade de Brasília, Brasília, 2014

[47] GASSEND, B. et al. “Silicon physical random functions” em: Proceedings of the 9th ACM Conference on Computer and Communications Security. New York, NY, USA: ACM, 2002. (CCS ’02), p. 148–160. ISBN 1-58113-612-9. Disponível em: http://doi.acm.org/10.1145/586110.586132

[48] SEDRA, S.; SMITH, K.. “Microeletrônica”. 4ª. Edição, Pearson Makron Books, São Paulo, Brasil, 2005.

[49] KATZENBEISSER, S. et al. “Pufs: Myth, fact or busted? a security evaluation of physically unclonable functions (pufs) cast in silicon” em: Cryptographic Hardware and Embedded Systems – CHES 2012. [S.l.]: Springer, 2012. p. 283–301.

[50] SUH, G. E.; DEVADAS, S. “Physical unclonable functions for device authentication and secret key generation” em: ACM. Proceedings of the 44th annual Design Automation Conference. [S.l.], 2007. p. 9 – 14.

[51] Noor N., Silva H. (2020) “Phase Change Memory for Physical Unclonable Functions” em: Suri M. (eds) Applications of Emerging Memory Technology. Springer Series in Advanced Microelectronics, vol 63. Springer, Singapore. Disponível em: https://doi.org/10.1007/978-981-13-8379-3_3

[52] QU, G.; YIN, C.-E. “Temperature-aware cooperative ring oscillator puf” em: IEEE. Hardware-Oriented Security and Trust, 2009. HOST’09. IEEE International Workshop on. [S.l.], 2009. p.36–42.

[53] Cabral I. “Oque são dapps? Apps descentralizados podem revolucionara Internet“ em: Techtudo, 2018. Disponível em: https://braziliex.com/blog/ethereum/voce-sabe-o-que-e-ethereum-dapp/ Acesso em: 8 maio. 2022.

[54] M. N. Islam and S. Kundu, "Remote Configuration of Integrated Circuit Features and Firmware Management via Smart Contract," 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 325-331, http://doi.org/10.1109/Blockchain.2019.00051.