UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Gabriel Estevam de Oliveira

# Proposta de Carimbo do Tempo Descentralizado e Preciso para a ICP-Brasil utilizando Sistemas Embarcados e Criptografia Pós-Quântica

Araranguá
2020

Gabriel Estevam de Oliveira

# Proposta de Carimbo do Tempo Descentralizado e Preciso para a ICP-Brasil utilizando Sistemas Embarcados e Criptografia Pós-Quântica

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.
Orientador: Prof. Martín Augusto Gagliotti Vigil, Dr.

Araranguá
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Oliveira, Gabriel Estevam
Proposta de Carimbo do Tempo Descentralizado e Preciso para a ICP-Brasil utilizando Sistemas Embarcados e Criptografia Pós-Quântica / Gabriel Estevam Oliveira ; orientador, Martín Augusto Gagliotti Vigil, 2020.
63 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2020.

Inclui referências.

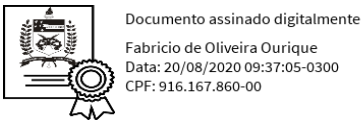
1. Engenharia de Computação. 2. Carimbo do Tempo. 3. Assinatura Digital. 4. Criptografia Pós-Quântica. 5. Sistemas Embarcados. I. Augusto Gagliotti Vigil, Martín. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Gabriel Estevam de Oliveira

**Proposta de Carimbo do Tempo Descentralizado e Preciso para a ICP-Brasil utilizando Sistemas Embarcados e Criptografia Pós-Quântica**

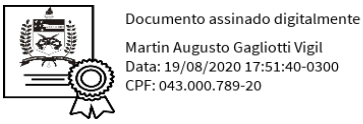
Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "Bacharel em Engenharia de Computação" e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 4 de agosto de 2020.

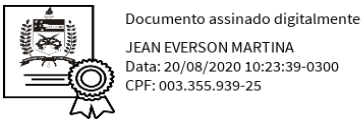



Prof. Fabrício de Oliveira Ourique, Dr.
Coordenador do Curso

**Banca Examinadora:**




Prof. Martín Augusto Gagliotti Vigil, Dr.
Orientador




Prof. Jean Everson Martina, Dr.
Avaliador


Roberto Alves Gallo Filho, Dr.
Avaliador

## AGRADECIMENTOS

Agradeço aos meus pais por possibilitarem eu estudar em tempo integral. Agradeço aos professores pela dedicação em ensinar, especialmente aos que tive a oportunidade de trabalhar extraclasse. E agradeço aos colegas pelas parcerias e incontáveis horas de estudos juntos.

## RESUMO

O modelo de Carimbo do Tempo aceito juridicamente no Brasil é o baseado em Autoridades de Carimbo do Tempo (ACT), regulamentadas pela Infraestrutura de Chaves Públicas Brasileira. Contudo, o modelo está sujeito a problemas causados pela centralização e pelo método de obtenção do carimbo. Além disso, aplicações como leilões online e mercado de ações necessitam de precisão de tempo superior ao fornecido pelo modelo. Diante disso, este trabalho propõe um dispositivo de carimbo do tempo compacto, de baixo custo e com criptografia pós-quântica com o objetivo de minimizar a centralização e aumentar a precisão de tempo. Por fim é apresentado uma prova de conceito e uma série de experimentos que mostram a factibilidade e eficácia do dispositivo.



**Palavras-chave**: Carimbo do Tempo. Autoridade de Carimbo do Tempo. Dispositivo. Centralização. Precisão.

## ABSTRACT

The time stamping scheme legally accepted in Brazil is based on Time Stamping Authority (TSA) and is regulated by Brazilian Public Key Infrastructure. In this scheme, concerns arise from system centralization and timestamps requests. Moreover, applications such as online auctions and stock markets need higher time precision than that the TSA-based schemes can provide. This work proposes a new, low cost, and compact time stamping device using post-quantum cryptography. Our goal is to minimize centralization and increase time precision in time stamping. Moreover, we provide a proof of concept and conduct experiments showing our proposal is feasibility and effective.



**Keywords**: Timestamp. Time Stamping Authority. Device. Centralization. Precision.

## LISTA DE FIGURAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AC | Autoridade Certificadora |
| AC-Raiz | Autoridade Certificadora Raiz |
| ACT | Autoridade de Carimbo do Tempo |
| DCT | Dispositivo de Carimbo do Tempo |
| DNS | *Domain Name System* |
| EAT | Entidade de Auditoria de Tempo |
| ETSI | *European Telecommunication Standard Institute* |
| GPS | *Global Positioning System* |
| HTTP | *Hypertext Transfer Protocol* |
| ICP-Brasil | Infraestrutura de Chaves Públicas Brasileira |
| IEEE | *Institute of Electrical and Electronics Engineers* |
| IETF | *Internet Engineering Task Force* |
| IoT | *Internet of Things* |
| IRTF | *Internet Research Task Force* |
| ITI | Instituto Nacional de Tecnologia da Informação |
| LabSEC | Laboratório de Segurança em Computação |
| NIC.BR | Núcleo de Informação e Coordenação do Ponto BR |
| NIST | *National Institute of Standards and Technology* |
| NTP | *Network Time Protocol* |
| NWG | *Network Working Group* |
| PoC | *Prova de Conceito* |
| PTP | *Precision Time Protocol* |
| RCT | Rede de Carimbo de Tempo |
| RFC | *Request For Comments* |
| RTC | *Real Time Clock* |
| SAS | Sistema de Auditoria e Sincronismo |
| SCT | Sistema de Carimbo do Tempo |
| SHA | *Secure Hash Algorithm* |
| TCC | Trabalho de Conclusão de Curso |
| TPM | *Trusted Platform Module* |
| UFSC | Universidade Federal de Santa Catarina |
| USB | *Universal Serial Bus* |
| UTC | *Universal Time Coordinated* |
| WLAN | *Wireless Local Area Network* |
| WOTS | *Winternitz One-Time Signature Scheme* |
| XMSS | *eXtended Merkle Signature Scheme* |

## LISTA DE SÍMBOLOS

| | |
|---|---|
| $h$ | Função de Hash Criptográfico |
| $m$ | Tamanho em bits do retorno da função de hash |
| $e$ | Entrada da função de hash |
| $y_e$ | Filho à esquerda na Árvore de Merkle |
| $y_d$ | Filho à direita na Árvore de Merkle |
| $w$ | Parâmetro de segurança do XMSS |
| $X$ | Chave privada XMSS |
| $x_i$ | Elemento $i$ da chave privada |
| $Y$ | Chave pública XMSS |
| $y_i$ | Elemento $i$ da chave pública |
| $\alpha$ | Assinatura XMSS |
| $\bar{x}$ | Média Amostral |
| $E$ | Margem de Erro |
| $\mu$ | Média Populacional |
| $z_{\alpha/2}$ | Escore correspondente a área de uma distribuição normal padrão |
| $\sigma$ | Desvio-padrão |
| $n$ | Tamanho da Amostra |

## SUMÁRIO

# 1 INTRODUÇÃO

Carimbo do tempo é a prova que uma informação digital existia em uma determinada data e hora. Os carimbos do tempo são documentos eletrônicos emitidos por Autoridades de Carimbo do Tempo (ACT). No Brasil as ACT são regulamentadas pela resolução Nº 111 da Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil). O documento contém uma representação compacta e única da informação digital e é assinado digitalmente pela ACT (ITI, 2017b). Segundo o Instituto Nacional de Tecnologia da Informação (ITI), o carimbo do tempo confere as propriedades de integridade e tempestividade a documentos digitais (ITI, 2020). Integridade garante que o documento não foi alterado e tempestividade estabelece data e hora confiáveis para a existência do documento (MENEZES; OORSCHOT; VANSTONE, 1996).

O uso do carimbo do tempo foi regulamentado pela ICP-Brasil em 2008, uma operação em conjunto com a BRy Tecnologia e o Laboratório de Segurança em Computação (LabSEC) da Universidade Federal de Santa Catarina (UFSC). A iniciativa permitiu inserir datação com validade legal em documentos assinados, trazendo um nível à mais de segurança além da certificação digital comum. A solução passou a ser utilizada por instituições públicas e privadas, como órgãos do judiciário, prefeituras, secretarias, laboratórios médicos, entre outros (ITI, 2008). Atualmente o carimbo do tempo é aplicado em atividades de registros de: apólices de seguro, direitos autorais, diplomas digitais, ponto eletrônico e inúmeras outras aplicações. O que implica em uma crescente migração de documentos do domínio análogico para o digital no cenário atual do Brasil (SANTIAGO, 2019).

Um caso em que é indispensável o carimbo do tempo é o projeto do Diploma Digital do Ministério da Educação. Em 2019, a UFSC foi a pioneira ao implantar um projeto-piloto para emissão de diplomas assinados digitalmente. Segundo Sergio Roberto de Lima e Silva Filho - consultor comercial da BRy Tecnologia - "O carimbo do tempo é a tecnologia que comprova a data e a hora que o documento foi emitido, e o certificado digital, a que garante a autenticidade do diploma". A BRy Tecnologia em conjunto com o LabSEC implementaram o projeto-piloto na UFSC (NSC, 2020).

Almeja-se no futuro que a tecnologia de carimbo do tempo seja difundida a ponto de tornar-se invisível ao usuário, estando presente em computadores, smartphones, câmeras de vigilância, automóveis, *Internet of Things* (IoT), entre outros. Com o crescimento das aplicações de IoT também aumentou-se a demanda por segurança, como relata Gabriel Jório - Superintendente de Operações e Comercial da Valid Certificadora. Como exemplo desta demanda, a certificação de bombas de combustíveis e relógios medidores de energia (CRYPTO, 2020).

A preocupação com a validade na datação de documentos é justificada pela facilidade de manipulação dos relógios dos computadores. De acordo com a ICP-Brasil é

facultativo a utilização de carimbo do tempo em documentos assinados eletronicamente, contudo irá depender das exigências jurídicas da aplicação. A ICP-Brasil possui um conjunto de documentos que regulamentam a geração e uso de carimbos do tempo. Contudo, cumprir todos os requisitos exigidos pela ICP-Brasil para se tornar uma ACT pode ser um processo dificultoso e caro. Existem apenas 8 ACT credenciadas em 2020 (ITI, 2017a) e apenas uma empresa que fornece Sistema de Carimbo do Tempo (SCT) homologado (ITI, 2017c), o que implica em uma alta centralização. Uma ACT é apta a conter mais de um SCT, podendo inclusive distribuir o serviço em outras empresas e instituições. Contudo, todos os SCT de uma ACT estão sob seu gerenciamento e mantêm os mesmos requisitos técnicos (CRYPTO, 2016).

A centralização do sistema de carimbo do tempo pode provocar problemas de indisponibilidade de serviço, escalabilidade, necessidade de infraestrutura avançada, entre outros (COULOURIS *et al.*, 2013). Além disso, o mundo vive um movimento rumo à descentralização de processos. Como exemplo as criptomoedas frente aos sistemas fiduciários tradicionais (CRYPTO, 2019). E em um cenário de descentralização, a validade jurídica é substituída por provas que podem ser evidenciadas pela comunidade através de métodos e protocolos auto-confiáveis (sem terceira parte confiável), como no caso do blockchain.

Outro problema deste modelo é a latência ou atraso de rede. A forma mais usual de obter um carimbo do tempo é através da Internet ou rede privada, porém implica em um atraso na marcação do tempo. Empresas e instituições podem possuir SCT locais às suas dependências (ICP-BRASIL, 2015c), mas a prática não é acessível à microempresas, trabalhadores autônomos e usuários comuns.

Segundo Kurose e Ross (2013), no melhor dos casos o atraso de rede é da ordem de milissegundos mas pode chegar a casa de centésimos ou décimos de segundo. No entanto, aplicações como leilões online e transações financeiras no mercado de ações necessitam de carimbos do tempo com precisão na ordem de milissegundos e algumas vezes microssegundos, como relata Broby, Basu e Arulselvan (2019).

Diante disso, este trabalho propõe um sistema de carimbo do tempo alternativo aos SCT das ACT. Com um hardware compacto, de baixo custo e com criptografia pós quântica com o objetivo de minimizar os problemas da centralização e aumentar a precisão de tempo.

O restante deste trabalho é organizado da seguinte forma. No capítulo 2 são apresentados conceitos básicos para o entendimento do trabalho. Na seção 3 são apresentados trabalhos relacionados. No capítulo 4 é apresentado o modelo de carimbo do tempo da ICP-Brasil. No capítulo 5 é apresentado o modelo proposto. No capítulo 6 é apresentado uma prova de conceito. No capítulo 7 é apresentado os experimentos realizados. E no capítulo 8 as considerações finais.

## 1.1 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste Trabalho de Conclusão de Curso (TCC).

### 1.1.1 Objetivo Geral

Propor uma solução de carimbo do tempo baseado no modelo da ICP-Brasil que permita difundir seu uso, amenizar o problema de centralização e aumentar a precisão na marcação do tempo.

### 1.1.2 Objetivos Específicos

- Estudar o modelo de carimbo do tempo da ICP-Brasil.
- Propor um modelo de carimbo do tempo alternativo.
- Realizar uma revisão bibliográfica sobre trabalhos relacionados.
- Construir uma prova de conceito para o modelo proposto.
- Analisar a factibilidade e eficácia da proposta através da prova de conceito.

## 1.2 METODOLOGIA

O estudo do modelo de carimbo do tempo da ICP-Brasil foi realizado através da revisão dos documentos regulamentadores DOC-ICP-11 (ICP-BRASIL, 2015c), DOC-ICP-12 (ICP-BRASIL, 2019), DOC-ICP-13 (ICP-BRASIL, 2015b) e DOC-ICP-14 (ICP-BRASIL, 2015a) disponibilizados pelo ITI.

A criação da proposta de um modelo alternativo de carimbo do tempo fez-se a partir de ideias para melhoramento do modelo atual. O processo contou com o estudo de tecnologias de sistemas embarcados, estudo de esquemas de assinatura digital e implementações para testes preliminares de comunicação e desempenho.

A revisão bibliográfica contou com uma busca por trabalhos relacionados através das palavras-chaves: *device*, *timestamp*, *time stamping authority* e seus sinônimos. A busca foi realizada nas seguintes plataformas: *Xplore Digital Library* do *Institute of Electrical and Electronics Engineers* (IEEE), *Springer Link*, *Science Direct*, *ACM Digital Library*, *Research Gate* e *Google Scholar*.

A prova de conceito para o modelo proposto foi realizada através da construção de um protótipo e de inquirição científica. Algumas premissas foram justificadas através da implementação e outras através de trabalho de pesquisa.

Por fim, realizou-se testes e experimentos para analisar a factibilidade e eficácia da proposta.

# 2 REFERENCIAL TEÓRICO

Nesta seção são apresentados conceitos importantes para este trabalho e que serão necessários para os próximos capítulos.

## 2.1 FUNÇÃO DE HASH CRIPTOGRÁFICO

Uma função de hash mapeia um conjunto de bits de tamanho arbitrário para um conjunto de bits de tamanho fixo. É definida por $h : \{0,1\}^* \rightarrow \{0,1\}^m$, sendo $m$ um inteiro positivo (BUCHMANN; KARATSIOLIS; WIESMAIER, 2013).

Funções de hash criptográfico possuem a propriedade de resistência a colisão por ser inviável encontrar $e$ e $e'$ distintos tal que $h(e) = h(e')$. Esta propriedade permite que as funções de hash criptográfico sejam utilizadas para verificação de integridade de documentos (BUCHMANN; KARATSIOLIS; WIESMAIER, 2013).

Um exemplo de função de hash criptográfico é a família de funções *Secure Hash Algorithm* (SHA) criadas pelo *National Institute of Standards and Technology* (NIST) dos Estados Unidos. A SHA possui 4 algoritmos criptográficos, que produzem hashes de 160 a 512 bits. Esses algoritmos criptográficos são considerados seguros pois garantem ser computacionalmente inviável: encontrar a entrada que produziu um determinado hash; e, encontrar duas entradas diferente que produzam o mesmo hash (NIST, 2002).

## 2.2 ÁRVORE DE MERKLE

Árvores de Merkle são árvores binárias que permitem verificar a integridade de um conjunto de documentos. Os hashes dos documentos são dispostos nas folhas da árvore e os nós internos - até a raiz - são construídos concatenando o filho à esquerda $y_e$ com o filho à direita $y_d$ e calculando o hash $h(y_e||y_d)$ (VIGIL *et al.*, 2015).

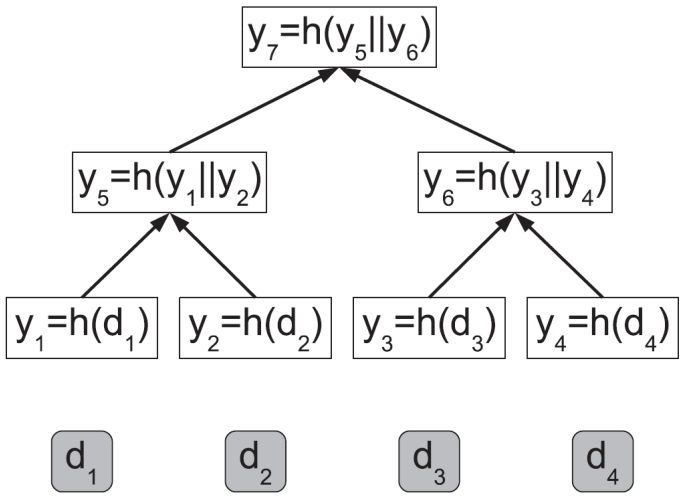
A Figura 1 mostra uma Árvore de Merkle.

Para verificar a integridade de um documento não é necessário conhecer todos os outros documentos, basta ter o caminho de autenticação. O caminho de autenticação é composto pelos nós irmãos do caminho da folha - onde se encontra o hash do documento - até a raiz da árvore. Isso totaliza $\lceil log2(n) \rceil$ hashes, sendo $n$ o número de documentos. Com o caminho de autenticação é possível reconstruir a raiz da árvore e comparar com a raiz original. Se forem iguais significa que a árvore contém o hash do documento.

## 2.3 ASSINATURA DIGITAL

A assinatura digital assegura integridade, autenticidade e não repúdio a documentos digitais. Integridade garante que o documento não foi alterado. Autenticidade permite identificar a origem do documento. E não-repúdio impede que o remetente refute a autoria do documento (VIGIL *et al.*, 2015).

Figura 1 – Árvore de Merkle.




Fonte: (VIGIL *et al.*, 2015)

Algoritmos de assinatura digital proveem 3 procedimentos: geração de chaves pública e privada, assinatura e verificação. A chave privada é utilizada para emitir a assinatura. E a chave pública é utilizada para a verificação da assinatura (VIGIL *et al.*, 2015). Um exemplo de algoritmo de assinatura digital é o RSA[1] que é baseado no problema de fatoração de números grandes (OLIVEIRA; FRANCO, 2017).

### 2.3.1 XMSS

O *eXtended Merkle Signature Scheme* (XMSS) é um algoritmo de assinatura digital baseado em função de hash, mais especificamente na propriedade de resistência a colisões das funções de hash, o que o torna um algoritmo resistente a ataques de computadores quânticos (IRTF, 2020).

O XMSS baseia-se no algoritmo *Winternitz One-Time Signature Scheme* (WOTS). No WOTS, cada par de chaves produz apenas uma assinatura. Para realizar o gerenciamento de múltiplas chaves o XMSS utiliza uma Árvore de Merkle. Desta forma, o WOTS aliado a uma Árvore de Merkle formam o XMSS.

#### 2.3.1.1 WOTS

O algoritmo WOTS utiliza a propriedade de resistência a colisões das funções de hash para produzir assinaturas digitais. Consiste em dividir a mensagem que se deseja assinar em segmentos menores e codificar cada segmento em um valor em uma base numérica escolhida. Para cada segmento é gerado um par de chaves pública e privada.

[1] RSA - Algoritmo criptográfico criado por Rivest, Shamir e Adleman (1978).

Para gerar a assinatura, a chave privada é submetida à função de hash repetidamente(realimentando o resultado) o número de vezes do valor codificado do segmento correspondente, produzindo um segmento assinado para cada segmento da mensagem original.

Para verificar a assinatura, cada segmento assinado é submetido à função de hash repetidamente(realimentando o resultado) o número de vezes do complemento na base numérica do valor codificado do segmento. Se os segmentos resultantes forem iguais as chaves públicas dos segmentos correspondentes então a assinatura estará correta.

Perin *et al.* (2018) definem o WOTS mais formalmente como segue. Dados $m$ o tamanho em bits do retorno de uma função de hash criptográfico $h$ e $w$ um parâmetro de escolha, com $w \in \mathbb{N}^*$ e $w > 1$:

$$t_1 = \left\lceil \frac{m}{w} \right\rceil, \quad t_2 = \left\lceil \frac{\lfloor log_2(t_1) \rfloor + 1 + w}{w} \right\rceil \quad \text{e} \quad t = t_1 + t_2 \,.$$

Por exemplo, para a função de hash criptográfico SHA256[2], que produz um resumo criptográfico com o tamanho de 256 bits, e $w$ igual a 4, recomendado por Perin *et al.* (2018), temos $t_1 = 64$, $t_2 = 3$ e $t = 67$.

Para a geração das chaves, escolhem-se números aleatórios $X = (x_{t-1}, \ldots, x_0)$ com cada $x_i$ de tamanho $m$. $X$ será a chave privada. Para encontrar a chave pública, $Y = (y_{t-1}, \ldots, y_0)$, aplica-se $h$ repetidamente(realimentado o resultado) um número de $2^w - 1$ vezes a cada elemento $x_i$, obtendo o $y_i$ correspondente.

Para gerar a assinatura aplica-se $h$ a mensagem e separa-se o resultado em $t_1$ segmentos de base $w$, obtendo $B_1 = (b_{t-1}, \ldots, b_{t-t_1})$. Então calcula $c$ como segue:

$$c = \sum_{i=t-t_1}^{t-1} 2^w - 1 - b_i.$$

Após isso divide-se $c$ em $t_2$ segmentos - preenchendo com zeros a esquerda caso necessário - obtendo $B_2 = (b_{t_2-1}, \ldots, b_0)$. Com $B = B_1 \cup B_2$, obtém-se a assinatura da seguinte forma:

$$\alpha = (h^{b_{t-1}}(x_{t-1}), ..., h^{b_0}(x_0)).$$

Para verificar a assinatura, calcula-se:

$$Y' = (h^{2^w-1-b_{t-1}}(\alpha_{t-1}), ..., h^{2^w-1-b_0}(\alpha_0)).$$

Se $Y' = Y$ então a assinatura está correta.

[2] Função de Hash Criptográfico recomendada por Cooper *et al.* (2019) para *Hash-Based Signature Scheme*, incluindo XMSS.

## 2.4 SISTEMAS EMBARCADOS

Segundo Li e Yao (2003), sistemas embarcados são sistemas computacionais que integram hardware e software projetados para realizar funções dedicadas. Sistemas embarcados geralmente são construídos utilizando microprocessadores de baixo custo, com baixo consumo de energia e baixo aquecimento. O software de um sistema embarcado é chamado de *firmware* e é especialmente projetado para obter o melhor desempenho com as especificações do hardware (LI; YAO, 2003).

### 2.4.1 Microcontrolador ESP32

Um microcontrolador é um dispositivo que integra um microprocessador e periféricos como memórias voláteis e não-voláteis, temporizadores, conversores analógico-digital, entre outros. O ESP32 é um microcontrolador que possui um *Real Time Clock* (RTC) interno, comunicação Wifi e comunicação serial do tipo *Universal Serial Bus* (USB).

Segundo a Espressif (2020):

> O ESP32 é um microcontrolador com conexão 802.11 b/g/n/e/i (2.4GHz) WLAN e Bluetooth 4.2, otimizado para baixo consumo de energia, eletrônicos de consumo móveis, sem fio e dispositivos de IoT. Integra todas as funcionalidade de WLAN e Bluetooth em um único equipamento de baixo custo, layout amigável para implementação e flexibilidade para plataformas customizáveis.

O microcontrolador também conta com o módulo *Cryptographic hardware acceleration* que realiza aceleração em hardware das funções criptográficas AES, SHA e RSA. Segundo a Espressif (2020) a aceleração em hardware permite executar operações significativamente mais rápido do que se fossem implementadas somente em software.

A Figura 2 mostra os componentes que compõem o microcontrolador ESP32.

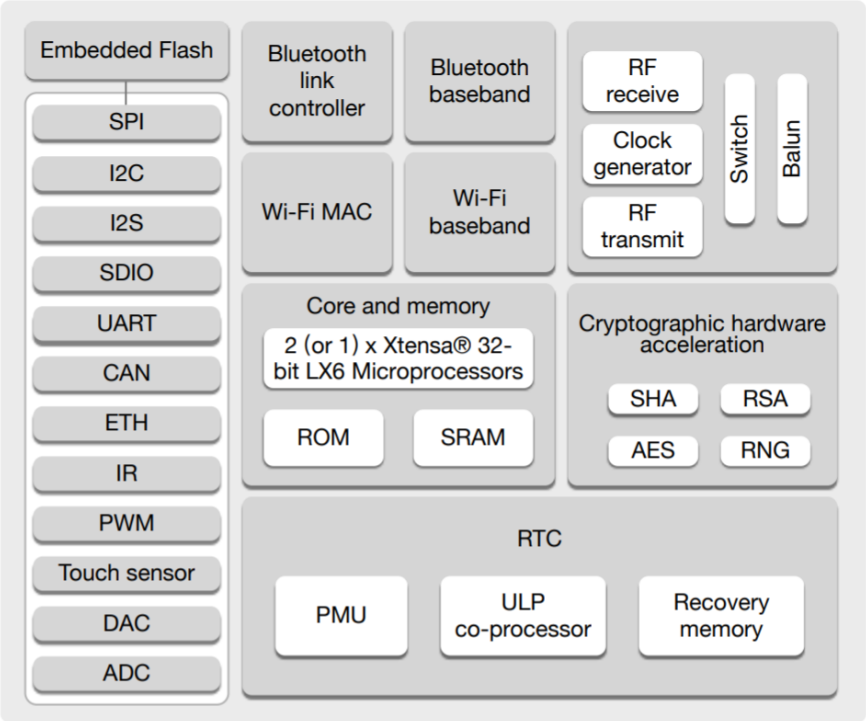
## 2.5 ESTATÍSTICA INFERENCIAL - ESTIMAÇÃO DA MÉDIA POPULACIONAL

A estatística inferencial é utilizada para fazer inferências ou tirar conclusões a partir de uma amostra populacional. Dentro da estatística inferencial existem dois principais parâmetros que permitem avaliar uma amostra, a proporção e a média populacional (TRIOLA, 2013). Mais a frente neste trabalho utilizará-se a média populacional para obter estimativas a partir de testes experimentais.

A média populacional é calculada de forma semelhante a média simples, o diferencial é a inserção de um intervalo de confiança. O intervalo de confiança é uma faixa de valores estabelecida em torno da média, com o objetivo de acrescentar um fator de qualidade à estimativa (TRIOLA, 2013).

Segundo Triola (2013) o intervalo de confiança para média populacional é definido pela segunda fórmula:

Figura 2 – Diagrama de Blocos Funcionais do ESP32.




Fonte: (ESPRESSIF, 2020)

$$\bar{x} - E < \mu < \bar{x} + E,$$

sendo $\bar{x}$ a média amostral, $\mu$ a média populacional e $E$ a margem de erro. A margem de erro é definida pela seguinte fórmula:

$$E = z_{\alpha/2}.\frac{\sigma}{\sqrt{n}},$$

sendo $z_{\alpha/2}$ o escore correspondente ao nível de confiança (determinado pela área de uma distribuição normal padrão, por exemplo $z_{\alpha/2} = 1.96$ para um nível de confiança de 95%), $\sigma$ o desvio padrão e $n$ o tamanho da amostra.

O tamanho da amostra escolhida influencia na margem de erro, assim como o desvio padrão. Para determinar o tamanho da amostra pode-se resolver a equação da margem de erro para $n$ como a seguir

$$n = \left[\frac{z_{\alpha/2}.\sigma}{E}\right]^2.$$

Como tipicamente o valor de $\sigma$ não é conhecido pode-se iniciar o processo de coleta da amostra e refinar os parâmetro da equação conforme desejável com novos dados amostrais.

# 3 TRABALHOS RELACIONADOS

Muitos trabalhos propuseram soluções alternativas de carimbos do tempo. A maioria destes trabalhos são focados na descentralização do modelo de carimbo do tempo. Como é o caso de (HARMANN, 2019) com a utilização de múltiplos servidores e verificação cruzada, (NEUMANN; HEEN; ONNO, 2014) com a utilização de servidores de DNS (*Domain Name System*) e diversos trabalhos utilizando blockchain como (GIPP; MEUSCHKE; GERNANDT, 2015). Contudo, em todos estes trabalhos os carimbos do tempo possuem acurácia de múltiplos segundos a minutos.

Em outra vertente, (KAKEI *et al.*, 2012) propuseram uma solução de carimbo do tempo *off-line* utilizando *Trusted Platform Module* (TPM). O TPM é um chip seguro, resistente a violação, montado diretamente na placa-mãe de um computador. Desta forma, o carimbo do tempo é gerado e assinado com criptografia RSA dentro do TPM, prevenindo a falsificação do tempo no carimbo. O TPM insere no carimbo um tempo relativo contado desde sua última atualização. A atualização é feita através de uma ACT via internet, que registra o tempo absoluto da atualização do TPM. Para obter o tempo absoluto do carimbo deve-se resgatar o tempo da correspondente atualização na ACT e somar com tempo fornecido pelo TPM. No entanto, esta abordagem utiliza a frequência do relógio do computador para a contagem relativa de tempo, que pode ser um ponto de vulnerabilidade. Além disso, necessita de uma adaptação de hardware para instalação do TPM. Ademais, os autores não revelaram dados de experimentos de acurácia, apenas que o carimbo do tempo leva pouco mais de um segundo para ser gerado.

Por fim, (STARNBERGER; FROIHOFER; GOESCHKA, 2010) propuseram um dispositivo de carimbo do tempo utilizando *smart cards* para leilões online. Um *smart card* é composto por um circuito integrado montado sobre um cartão plástico, capaz de processar e armazenar dados. A proposta se preocupa com a centralização e possíveis ataques em leilões online. No modelo proposto, o *smart card* utiliza um protocolo seguro de sincronização de tempo e emite uma assinatura que é gerada dentro do dispositivo, semelhante ao caso anterior. Mas nesse caso, o *smart card* possui um oscilador próprio, independente do computador a qual é conectado. No entanto, a baixa capacidade de processamento dos *smart cards* foi um fator impeditivo para a sincronização do relógio e precisão do tempo.

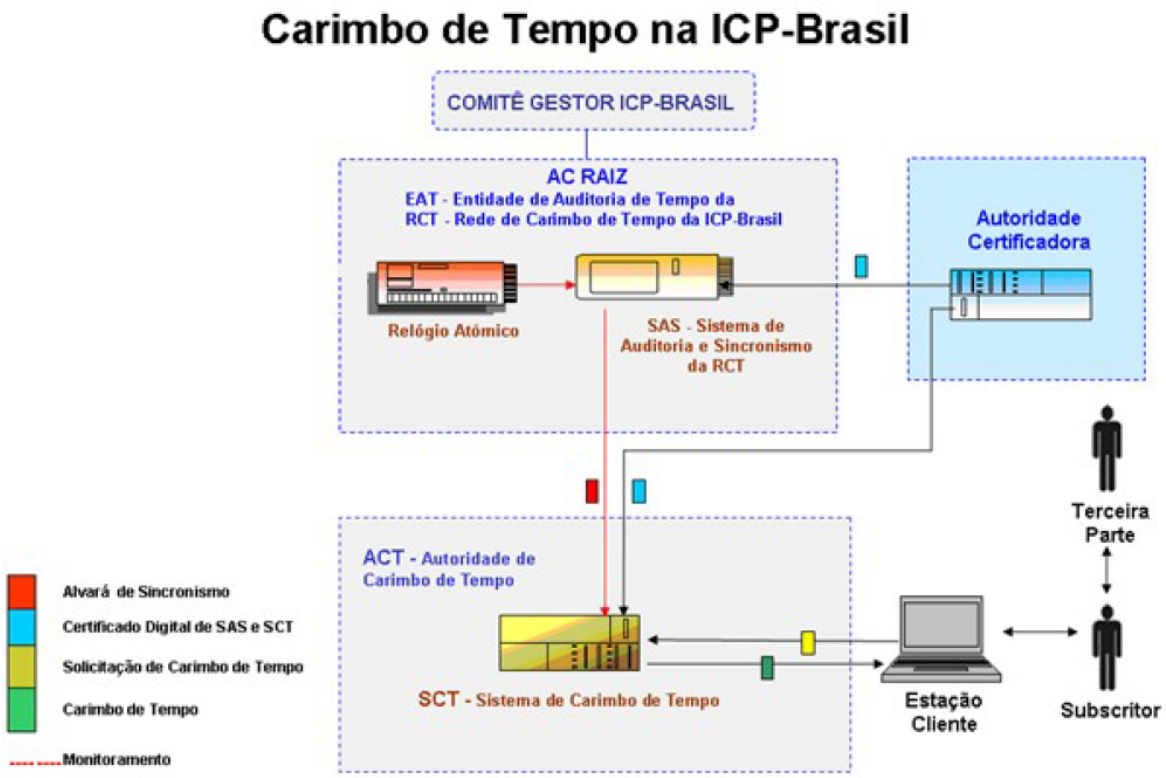
# 4 MODELO DE CARIMBO DO TEMPO DA ICP-BRASIL

A ICP-Brasil possui um conjunto de documentos que regulamentam a geração e uso de carimbos do tempo no Brasil. O ICP-Brasil (2015c) traz uma visão geral do sistema de carimbos do tempo da ICP-Brasil. O ICP-Brasil (2015c) declara os requisitos mínimos para as declarações de práticas das ACT. O ICP-Brasil (2015b) declara os requisitos mínimos para as políticas de carimbo do tempo. E o ICP-Brasil (2015a) descreve os procedimentos para auditoria do tempo. Todos os documentos seguem as normas do NWG (2003), do NWG (2001) e do ETSI (2011).

## 4.1 VISÃO GERAL

A Figura 3 representa uma visão geral da estrutura de carimbo do tempo da ICP-Brasil.

Figura 3 – Modelo de Funcionamento do Carimbo do Tempo da ICP-Brasil.




Fonte: (ICP-BRASIL, 2015c)

O Comitê Gestor da ICP-Brasil é responsável pelas normas e implantação do modelo. A Autoridade Certificadora Raiz (AC-Raiz) credencia, fiscaliza e audita entidades da ICP-Brasil e atua como Entidade de Auditoria de Tempo (EAT). As Autoridades

Certificadoras (AC) são responsáveis por emitir, renovar e revogar os certificados dos SCT e do Sistema de Auditoria e Sincronismo (SAS) da AC-Raiz.

As ACT são as entidades responsáveis pela emissão dos carimbos do tempo. As ACT devem operar um ou mais SCT, conectados à Rede de Carimbo de Tempo (RCT). O Subscritor é o cliente pessoa física ou jurídica que solicita o carimbo do tempo. E a Terceira Parte é a pessoa ou entidade na qual é apresentado o carimbo do tempo, podendo verificar sua validade.

## 4.2 SINCRONIZAÇÃO DO TEMPO

O modelo utiliza um mecanismo para garantir o sincronismo dos relógios e a rastreabilidade do tempo nos equipamentos das entidades que compõem a estrutura de carimbo do tempo da ICP-Brasil, incluindo as ACT. O relógio atômico da ICP-Brasil fornece a hora *Universal Time Coordinated* (UTC)[1] para o SAS da AC-Raiz. E o SAS dissemina a hora para os equipamentos das ACT e emite o alvará de sincronismo.

## 4.3 OBTENÇÃO DO CARIMBO DO TEMPO

A ICP-Brasil define duas formas de obtenção do carimbo do tempo. A primeira é a solicitação presencial através da entrega de uma mídia física diretamente nas dependências da ACT. E a segunda forma é por solicitação remota através de um serviço disponibilizado pela ACT por meio de uma rede privada ou pela Internet.

## 4.4 ASPECTOS DE SEGURANÇA DAS ACT

A ICP-Brasil exige que as ACT utilizem um Módulo de Segurança Criptográfico para a geração de chaves criptográficas e assinatura digital. Além disso é exigido um rígido controle de segurança física, procedimental e de pessoal aos SCT. Incluem-se níveis de acesso físico, sistemas de detecção, normas de armazenamento de dados, qualificação de pessoal, restrições de acesso, etc.

## 4.5 PROBLEMAS DO MODELO

O modelo de carimbo do tempo da ICP-Brasil apresenta, sem dúvidas, altíssimo nível de segurança. Contudo, o modelo implica em limitações como centralização e atraso na marcação do tempo.

A centralização é ocasionada pelo número reduzido de ACT e pelo custo elevado para dispor de um SCT. Entre os problemas que podem ser causados pela centralização estão: indisponibilidade de serviço, escalabilidade, necessidade de infraestrutura avançada, etc (COULOURIS *et al.*, 2013).

[1] Tempo decorrido em segundos desde 1 de Janeiro de 1970.

E o atraso na marcação do tempo é devido aos métodos de obtenção do carimbo do tempo. Mesmo quando a solicitação é feita via Internet o atraso no melhor dos casos é na ordem de milissegundos, mas pode chegar a centésimos ou décimos de segundo (KUROSE; ROSS, 2013). Isso também impede aumentar a precisão do tempo inserido no carimbo, para milissegundo por exemplo, pois o atraso de comunicação deve ser menor que a precisão utilizada.

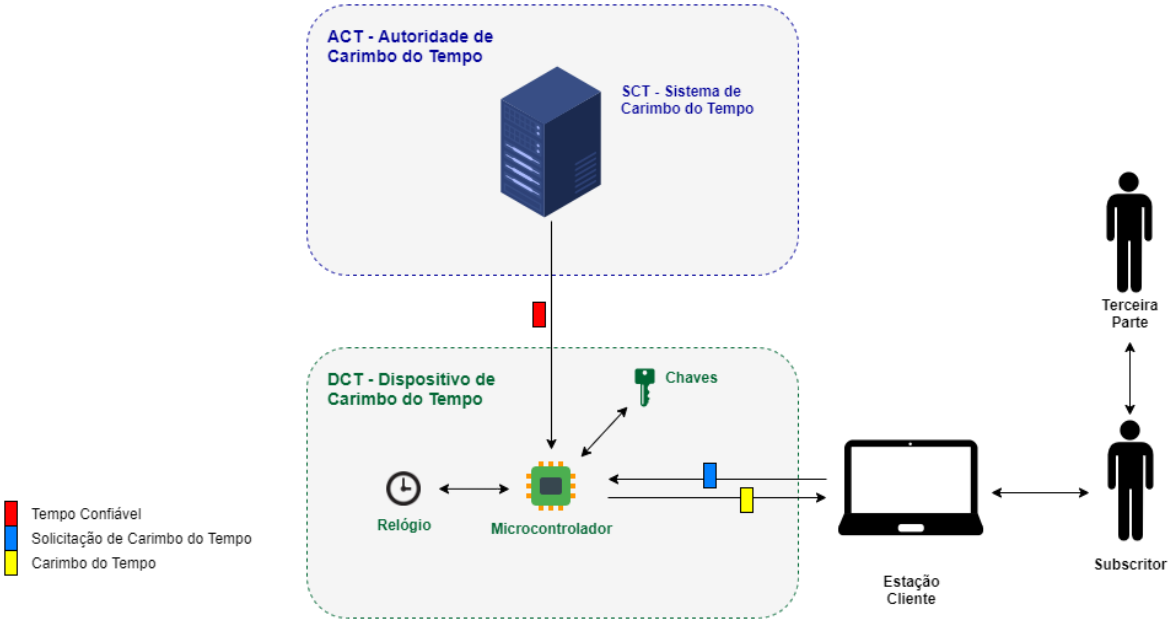
# 5 MODELO PROPOSTO - DISPOSITIVO DE CARIMBO DO TEMPO

Nesta seção é apresentada uma proposta de modificação no modelo de carimbo do tempo da ICP-Brasil.

## 5.1 VISÃO GERAL

A proposta consiste em adicionar um novo componente ao modelo. Este componente é o Dispositivo de Carimbo do Tempo (DCT). Na Figura 4 é representado a modificação na estrutura do modelo.

Figura 4 – Modelo Proposto.




Fonte: Elaborado pelo Autor

O DCT é um equipamento capaz de emitir carimbo do tempo e tem a finalidade de ser uma extensão dos SCT das ACT. Neste modelo a geração e assinatura do carimbo do tempo é feita dentro do DCT, que é de posse do subscritor. O DCT possui um microcontrolador, um relógio interno e um conjunto de chaves públicas e privadas. As chaves privadas são geradas dentro do microcontrolador e não serão acessíveis por nenhuma entidade em nenhum momento. O microcontrolador ajusta o tempo do relógio com base no tempo fornecido pelo servidor de tempo da ACT e emite o certificados de carimbo do tempo assinados. É de responsabilidade das ACT: homologar, auditar, certificar, sincronizar e eventualmente comercializar os DCT.

## 5.2 CARACTERÍSTICAS TÉCNICAS

O DCT deve possuir um tamanho compacto, menor que um cartão de crédito, o que o permite ser portável ao usuário. São restrições técnicas para o DCT:

- Possuir um relógio de tempo real (RTC).
- Possuir interface de comunicação de rede.
- Possuir interface de comunicação serial USB.
- Ser capaz de gerar e armazenar as chaves criptográficas.
- Ser capaz de executar um algoritmo de assinatura digital.
- Possuir alimentação interna.

## 5.3 SINCRONIZAÇÃO DO TEMPO

O relógio interno do DCT deve ser sincronizado periodicamente por um servidor de tempo da ACT. A sincronização acontecerá via Internet. Se o DCT perder a conexão com a Internet, depois de um período de tempo determinado pela ACT a sincronização perderá sua validade e o dispositivo deixará de emitir carimbos do tempo até uma nova sincronização. Para realizar a sincronização do relógio, deve-se estabelecer um canal de comunicação entre a ACT e o DCT que garanta a autenticidade das informações enviadas.

A sincronização do relógio segue o *Network Time Protocol* (NTP) estabelecido pelo NIC.BR (2020). No esquema define-se uma troca de mensagens para que o cliente descubra o deslocamento (*offset*) do seu tempo em relação ao servidor.

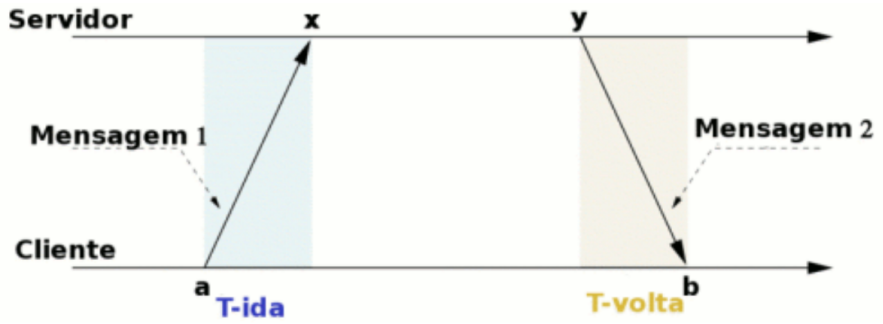
Segundo o NIC.BR (2020) a troca de mensagens tem a seguinte forma (vide Figura 5):

- O cliente marca o seu tempo atual **a**.
- O cliente envia a Mensagem 1 ao servidor com o tempo **a**.
- O servidor recebe a mensagem e marca o tempo em que recebeu como **x**.
- O servidor envia a Mensagem 2 com **a**, **x** e seu tempo atual **y**.
- O cliente recebe a mensagem com **a**, **x** e **y** e marca seu tempo atual **b**.

Por simplificação o esquema considera o tempo de ida da mensagem igual ao tempo de volta. Desta forma, o atraso (*delay*) da mensagem é definido como:

$$delay = \frac{(x - a) + (b - y)}{2}.$$

Figura 5 – Troca de Mensagens.




Fonte: (NIC.BR, 2020)

E o deslocamento é definido da seguinte forma:

$$offset = x - (a + delay) = \frac{x - a + y - b}{2}.$$

O *offset* é utilizado para ajustar do relógio local. Se o *offset* for positivo o relógio local está atrasado. Se o *offset* for negativo o relógio local está adiantado.

## 5.4 OBTENÇÃO DO CARIMBO DO TEMPO

Neste modelo é estabelecida uma nova forma de obtenção do carimbo do tempo. Consiste em o usuário conectar o DCT a uma porta USB de um computador e utilizar uma aplicação local para realizar a emissão do carimbo do tempo. Desta forma, o carimbo do tempo é gerado localmente.

## 5.5 ASPECTOS DE SEGURANÇA DO DCT

Para garantir a integridade e tempestividade nos carimbos de tempo emitidos pelos DCT são necessárias medidas que previnam eventuais vulnerabilidades de segurança do dispositivo. Por isso, nesta seção é descrito um modelo de ameaça para o DCT e algumas medidas que podem ser tomadas para corrigir ou amenizar as vulnerabilidades de segurança.

### 5.5.1 Modelo de Ameaças

O modelo de ameaças é um estudo do cenário, ambiente, contexto e circunstâncias que o sistema em questão está ou pode ser submetido. Tem o objetivo de levantar possíveis vulnerabilidades de segurança do sistema e prever ataques (SHOSTACK, 2014).

No modelo proposto o DCT é de posse do subscritor, que é um usuário no mundo real. Este usuário pode levar o DCT para qualquer lugar e aplicar qualquer técnica que

tenha disponível para tentar violar a segurança do dispositivo. Sendo que os dois principais pontos levantados como possiveis vulnerabilidades de segurança do DCT são:

- Adulteração de dados ou descoberta das chaves criptográficas por acesso a memória interna.
- Avanço ou retrocesso do RTC.

Quanto ao avanço ou retrocesso do RTC do dispositivo, tem-se conhecimento de técnicas que podem manipular a frequência do relógio. A frequência do relógio é suscetível a temperatura, tensão de alimentação e radiação eletromagnética. E quanto a adulteração ou descoberta de dados por acesso a memória interna pode ser realizada lendo os sinais diretamente nos pinos do microcontrolador com equipamento adequado.

No entanto, neste trabalho impede-se o subscritor de explorar as vulnerabilidades de acesso à memória, conforme medidas de segurança mencionadas na seção a seguir. Quanto às vulnerabilidades de avanço ou retrocesso de relógio, assume-se aqui que elas não são exploradas. Meios para impedi-las efetivamente são deixados para trabalhos futuros.

### 5.5.2 Medidas de Segurança

A ICP-Brasil define requisitos para a homologação de *tokens* criptográficos em um de seus manuais de condutas técnicas. *Tokens* criptográficos são hardwares com conexão USB, com capacidade para gerar e armazenar chaves criptográficas e realizar processamento criptográfico (ICP-BRASIL, 2017). Diferente dos DCT, os *tokens* criptográficos não possuem RTC. Contudo, podem compartilhar os requisitos técnicos de segurança.

Dentre os requisitos estabelecidos definem-se restringir acesso físico aos circuitos integrados com a finalidade de deter a observação, sondagem, manipulação e a substituição ou remoção de componentes do módulo. Para isso, o circuito integrado do módulo deve ser protegido por um invólucro que evidencie sinais de tentativas de violação (ICP-BRASIL, 2017).

Além dos requisitos já estabelecidos deve-se também acrescentar a blindagem térmica e eletromagnética, com o intuito de prevenir o ataque à manipulação da frequência do RTC do dispositivo. Atender tais requisitos são trabalhos futuros.

# 6 PROVA DE CONCEITO

Nesta seção será apresentada uma Prova de Conceito (PoC - *Proof of Concept*) para o modelo proposto. A PoC consistiu em analisar a viabilidade técnica do modelo quanto aos critérios de implementação e segurança. O critério de implementação avalia se é possível cumprir as funcionalidades esperadas para o DCT por meio do desenvolvimento de hardware e software. E o critério de segurança avalia se é possível alcançar os requisitos de segurança necessários para o DCT.

A PoC apresentada não tem com objetivo atender todos os requisitos estabelecidos no modelo, apenas mostrar que é possível construir um DCT. Portanto, foram assumidas algumas simplificações sem comprometer resultado da análise. O trabalho a seguir conta com o desenvolvimento de um dispositivo de carimbo do tempo, a implementação do firmware do dispositivo, a implementação da aplicação que solicita a emissão do carimbo do tempo e um servidor de tempo para sincronização do relógio do DCT.

## 6.1 DISPOSITIVO DE CARIMBO DO TEMPO

O microcontrolador utilizado para a construção do DCT foi o ESP32 na versão de desenvolvimento, apresentada na seção 2.4.1. O microcontrolador foi escolhido por possuir um RTC interno, comunicação serial USB, comunicação Wifi e tamanho compacto. O ESP32 na sua versão de desenvolvimento já possui conexão USB e antena Wifi, portanto não foi necessário acoplar componentes externos. Para a PoC, considerou-se desnecessário a utilização de alimentação interna. Durante a realização dos testes a alimentação foi suprida pela conexão USB.

Para o invólucro do dispositivo propõe-se o encapsulamento com resina epóxi. A resina epóxi possui uma composição química que proporciona alta adesão, resistência mecânica elevada, resistência a altas temperaturas e baixa absorção de umidade (FAN; WONG, 2001). Esta técnica já é utilizada para encapsulamento de circuitos integrados, como mostra Hadizadeh *et al.* (2019), Yamoaka, Kusuhara e Okabe (1988), Tao *et al.* (2007) e Okuno, Fujita e Ishikana (1999).

## 6.2 FIRMWARE

O firmware do dispositivo foi implementado em linguagem C[1] e incorpora bibliotecas específicas para o microcontrolador. As bibliotecas permitem realizar a comunicação Wifi e utilizar as funções criptográficas que contam com aceleração em hardware.

[1] Linguagem de programação de médio nível consolidada e altamente difundida na programação de microcontroladores (SCHILDT, 1996).

### 6.2.1 Funcionalidades

O firmware conta com as funcionalidades de emitir carimbo do tempo, consultar chave pública do dispositivo e alterar nome e senha da rede Wifi. A seguir é descrito cada uma delas:

- **Emitir carimbo do tempo**: permite solicitar a emissão de carimbo do tempo. Anexo à solicitação é submetido o hash do documento que se deseja carimbar.
- **Consultar chave pública**: permite consultar a chave que identifica unicamente o dispositivo. A chave pública permite ao Subscritor e a Terceira Parte consultar se o DCT consta na lista de DCT certificados ou revogados da ACT.
- **Alterar nome e senha da rede Wifi**: permite informar o nome e senha da rede Wifi que o DCT irá utilizar para sincronização do relógio. A conexão com a rede Wifi é ativada apenas no momento da sincronização.
- **Consultar status da rede**: informa se é possível estabelecer uma conexão de rede com o nome da rede e senha atuais.

### 6.2.2 Sincronização do Relógio

A sincronização do relógio é realizada periodicamente, caso seja possível conectar-se a rede Wifi informada e ao servidor de tempo da ACT. O procedimento de sincronização é descrito na seção 4.3. Quando o procedimento de sincronização é realizado com sucesso, a validade da sincronização do relógio do DCT é renovada por um período de tempo estabelecido pela ACT. A escolha do período de validade é determinada pela acurácia do relógio do DCT. A acurácia indica o desvio que a hora do relógio pode ter ao longo do tempo.

### 6.2.3 Certificado de Carimbo do Tempo

O certificado de carimbo do tempo é o documento emitido pelo DCT. O certificado utilizado neste trabalho não segue um modelo padronizado, e.g. X.509. O modelo de certificado criado tem apenas o objetivo de atender a PoC. O documento possui os seguintes atributos:

- **Hash do dado** (*DataHash*): identificação do documento carimbado.
- **Data e hora** (*Timestamp*): marcação temporal no momento da solicitação.
- **Chave pública** (*PublicKey*): permite verificar a assinatura do carimbo.
- **ACT** (*AuditEntity*): nome da ACT responsável pelo DCT.

- **Última sincronização** (*LastSync*): data e hora da última sincronização do relógio.
- **Servidor de tempo** (*TimeServer*): endereço do servidor de tempo.

O certificado de carimbo do tempo é retornado pelo DCT por comunicação serial após a solicitação de carimbo do tempo. O modelo de certificado de carimbo do tempo pode ser visto na Figura 6.

Figura 6 – Certificado de Carimbo do Tempo.

```
### BEGIN CERTIFICATE ###
DataHash:ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb
Timestamp: Wednesday, 26 Feb 2020 22:41:17:545
PublicKey: 2
961b6dd3ede3cb8ecbaacbd68de040cd78eb2ed5889130cceb4c49268ea4d506
9834876dcfb05cb167a5c24953eba58c4ac89b1adf57f28f2f9d09af107ee8f0
61be55a8e2f6b4e172338bddf184d6dbee29c98853e0a0485ecee7f27b9af0b4
AuditEntity: empty
LastSync: Wednesday, 26 Feb 2020 22:36:38:511
TimeServer: 10.1.1.13
#### END CERTIFICATE ####
```

Fonte: Elaborado pelo Autor

### 6.2.4 Assinatura

A assinatura é um conjunto de hashes que são retornados junto com o certificado de carimbo do tempo. O método utilizado para a assinatura do certificado é o XMSS, descrito na seção 2.3.1. A assinatura é única para cada certificado e pode ser verificada como demonstrado na seção 2.3.1.1. A função de hash utilizada para a compactação do certificado e em todas as operações do XMSS é a SHA256, como recomendado por Cooper *et al.* (2019) para esquemas de assinatura digital baseados em funções de hash, incluindo XMSS. Para o parâmetro $w$ foi adotado 4, como recomendado por Perin *et al.* (2018).

A escolha da função de hash utilizada e do parâmetro $w$ do XMSS implicam no tamanho das chaves e no número de vezes que é aplicado a função de hash durante a assinatura. Para a função SHA256 e $w$ com valor 4 então $t = 67$ (Vide seção 2.3.1.1), que é o número de hashes que compõem as chaves pública e privada. Com isso, no pior caso são aplicadas $67x(2^4 - 1) = 1005$ vezes a função de hash durante a assinatura.

O esquema de assinatura XMSS utiliza apenas uma vez cada chave, e cada chave privada possui um tamanho de $67x256 = 17152$ bits. As chaves são criadas dentro do DCT no momento em que é iniciado. Cada uma das chaves são compostas por 67 hashes, contudo para as chaves públicas pode-se criar uma Árvore de Merkle inserindo cada um dos 67 hashes nas folhas da árvore. Ao armazenar apenas a raiz da árvore diminui-se o

consumo de memória sem perder a integridade da chave. Cada uma das chaves públicas será representada por um único hash. E então uma nova árvore é criada tendo como suas folhas os hashes que representam as chaves públicas. A raiz dessa árvore é a chave pública do dispositivo. As chaves privadas não ficam armazenadas no dispositivo, apenas um número aleatório (semente) que permite obter as chaves. Com a semente e um fator de entropia é possível reconstruir as chaves em tempo de execução.

Essa abordagem permite obter um número maior de chaves. A implementação permitiu armazenar até 512 chaves, o que possibilita realizar 512 assinaturas. Para obter um número maior de assinaturas é necessário um procedimento de renovação das chaves.

Com esse esquema, cada certificado gerado contém o hash que representa a chave pública correspondente mais o caminho da árvore que permite reconstruir a chave pública do dispositivo. Para o conjunto de 512 chaves, o caminho possui o tamanho de 10 hashes.

## 6.3 APLICAÇÃO

A aplicação é um programa de computador escrito em linguagem Python 3[2]. Permite conectar-se ao dispositivo e realizar as operações descritas na seção 6.2.1 via comandos no terminal. O programa chama-se *app_ crypto_timestamp.py* e possui os seguintes comandos:

- **Solicitar carimbo de tempo**:

  Sintaxe: *python app_crypto_timestamp.py –port PORTA –certificate HASH*

- **Editar o nome da rede**:

  Sintaxe: *python app_crypto_timestamp.py –port PORTA –networkSSID NOME_DA_REDE*

- **Editar o senha da rede**:

  Sintaxe: *python app_crypto_timestamp.py –port PORTA –networkPassword SENHA*

- **Consultar status da rede**:

  Sintaxe: *python app_crypto_timestamp.py –port PORTA –networkStatus*

- **Consultar chave pública**:

  Sintaxe: *python app_crypto_timestamp.py –port PORTA –publicKey*

Foi implementado também na aplicação o esquema de assinatura XMSS. Isso permite que a aplicação verifique a assinatura e a autenticidade da chave do dispositivo. A autenticidade da chave garante que a assinatura foi gerada pelo dispositivo.

[2] Linguagem de programação de alto nível. Vide Python Software Foundation (2020).

## 6.4 SERVIDOR DE TEMPO

Um servidor de tempo foi implementado em NodeJS[3] para fornecer data e hora confiáveis para a sincronização do relógio do DCT. No modelo proposto a responsabilidade pelo servidor de tempo é da ACT. O DCT se conecta ao servidor de tempo por comunicação HTTP[4]. O método de sincronização foi descrito na seção 5.3. A resposta do servidor de tempo é assinada via XMSS e é verificada dentro do DCT, garantindo autenticidade na comunicação.

[3] Tecnologia que permite criar servidores web. Vide OpenJS (2020).

[4] *Hypertext Transfer Protocol* (HTTP) - Protocolo de comunicação cliente-servidor (MDN, 2020)

# 7 ANÁLISE DOS RESULTADOS

Neste capítulo são apresentados alguns experimentos realizados para análise do protótipo do DCT desenvolvido. Foram realizados experimentos de funcionamento, latência, tempo de resposta, precisão, sincronização e acurácia.

Os equipamentos utilizados nos experimentos foram: um ESP-WROOM-32 (daqui pra frente chamado de dispositivo); um computador Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz-3.0GHz, 16 GB de memória RAM e Sistema Operacional Linux Ubuntu 18.04.4 LTS x64; um computador Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz-2.70GHz, 8 GB de memória RAM e Sistema Operacional Windows 10 x64; e um *switch*[1] Intelbras SF 800Q. Nos experimentos que demandaram apenas 1 computador foi utilizado o computador com sistema operacional Linux.

Faz-se uma observação aqui que este não é o cenário mais ideal para realização dos testes. A utilização de sistemas operacionais de propósito geral pode acentuar os tempos obtidos nos experimentos, devido ao escalonador de processos desses sistemas. Para melhorar a acurácia dos resultados pode-se utilizar sistemas de tempo real para obtenção das amostras e a utilização de um analisador lógico diretamente nos pinos do microcontrolador.

Nos experimentos onde houve coleta de dados utilizou-se a estimativa de média populacional apresentada na seção 2.5. O cálculo da margem de erro indica o intervalo de confiança e também sugere se o tamanho da amostra foi suficiente. Um valor de erro proporcionalmente grande se comparado ao valor da média aponta que o tamanho da amostra não foi suficiente. Em todos os experimentos o parâmetro de confiança adotado foi de 95%.

## 7.1 FUNCIONAMENTO

Nesta seção é demonstrado a utilização de cada uma das funcionalidades através da aplicação descrita na seção 6.3. As funcionalidades do DCT foram detalhadas na seção 6.2.1.

Assim que o DCT é iniciado ele executa o procedimento para geração das chaves, uma luz no dispositivo indicará que ele está pronto para utilização. Após isso já é possível consultar a chave pública do dispositivo, como pode ser visto na Figura 7.

A seguir deve ser realizado a alteração do nome e senha da rede. A alteração do nome da rede pode ser vista na Figura 8 e a alteração da senha da rede pode ser vista na Figura 9.

Na sequência pode-se verificar o *status* da rede (Figura 10). Neste teste o dispositivo tenta conectar-se à rede informada anteriormente. Se for possível conectar-se à rede então

[1] Equipamento que permite comunicação de dispositivos em redes de computadores.

Figura 7 – Consultando a chave pública do DCT.

```
C:\crypto_timestamp_app>python app_crypto_timestamp.py --port COM6 --publicKey
PublicKey: f2eccca8144630bdf7f20451781c6f4b42aeecbd3ecd843616af16df97cfc65f
```

Fonte: Elaborado pelo Autor

Figura 8 – Alterando o nome da rede no DCT.

```
C:\crypto_timestamp_app>python app_crypto_timestamp.py --port COM6 --networkSSI
D "DESKTOP-TN0OAP8 9920"
New network ssid: DESKTOP-TN0OAP8 9920
```

Fonte: Elaborado pelo Autor

Figura 9 – Alterando a senha da rede no DCT.

```
C:\crypto_timestamp_app>python app_crypto_timestamp.py --port COM6 --networkPas
sword 1=R4c033
New network password.
```

Fonte: Elaborado pelo Autor

o dispositivo irá sincronizar seu relógio interno e estará disponível para emitir carimbos do tempo.

Figura 10 – Verificando o *status* da rede

```
C:\crypto_timestamp_app>python app_crypto_timestamp.py --port COM6 --networkSta
tus
Network available
```

Fonte: Elaborado pelo Autor

Finalmente é possível solicitar a emissão de carimbo do tempo, como pode ser visto na Figura 11. A assinatura também é retornada logo após o certificado, são 67 hashes de 256 bits. Na figura é mostrado apenas o início da assinatura.

## 7.2 LATÊNCIA

Um experimento de latência realizado aferiu o tempo despendido na comunicação entre a aplicação e o dispositivo. O tempo despendido indica o atraso entre a solicitação

Figura 11 – Emissão de certificado

```
C:\crypto_timestamp_app>python app_crypto_timestamp.py --port COM6 --certificat
e 0xca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb
Certificado:
#### BEGIN CERTIFICATE ####
DataHash:0xca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb
Timestamp:Thursday, March 26 2020 01:48:00:750
PublicKey:0
3a2984b853469bef3a9192b7785cc5d353c02a75289684b0581e09155a273593
f6835195f8fb92df09914fe75bdf816fd71212d27359e24d768af02f7bf0cc12
0ea8d864a26d487716e8c4d2e5b9a45198acb3bc6df8673c551cd1bb58dfe212
701a5567f13937e575d3f9ee5a393b6ac2c4f68ad1e49603da562ce23ba9231a
ae60f4f844448226fb9da3267cf017c2302b90f58c7499dba7b607baace368de
efaea838f1bacee092eb16e5ad6964163db13c20ecb9a7945f1ac5fa42d60871
AuditEntity:empty
LastSync:Thursday, March 26 2020 01:47:04:438
TimeServer:10.1.1.13
##### END CERTIFICATE #####
Assinatura:
ec29da661d83407b55a450e630c9c5d092b06c61068d08f9be65edfc065b8722
bb999a5aa7a935aa410ead5b1ae808f7473ae18554e0ef09bdace0bbeb952f56
b02579d41b2613c533bc0dc2e86d5ed3c58de106a5f6151698f89cfcaf62ff01
3356b2018067e002c3730aace10f3d10b88293d5437e6b84aa660044b84f1a7c
41bc6baf7275240dfb31b836c86babfcb5598ba838e1555b84c19fe698cb7085
24a3266a2b813b809270ca9e81c9a794d35ace96949f61a86a3479c5821ff7c0
```

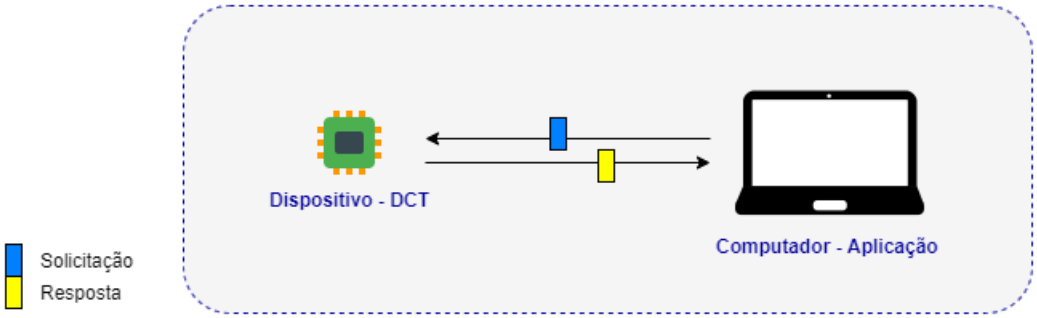
Fonte: Elaborado pelo Autor

de carimbo do tempo e a marcação do tempo pelo DCT. O experimento consiste em medir o tempo necessário para uma mensagem ser enviada de um computador até o dispositivo. O computador envia a mensagem e marca o tempo em que enviou, o dispositivo recebe a mensagem e envia uma resposta, o computador recebe a resposta e marca o tempo em que recebeu. O tempo entre o envio e o recebimento é o tempo de ida e volta da mensagem. Considerando que o tempo de ida e de volta são aproximadamente iguais, metade do valor obtido é o tempo que a mensagem levou para ser enviada do computador até o dispositivo. Essa abordagem permite aferir o tempo despendido na comunicação sem a necessidade de os relógios do computador e do dispositivo estarem sincronizados.

A Figura 12 ilustra o cenário do experimento Computador-Dispositivo.

Realizou-se também um experimento para medir o tempo necessário para comunicação entre dois computadores utilizando comunicação HTTP. O objetivo deste experimento foi montar um cenário o mais ideal possível para aferir o tempo mínimo despendido na comunicação HTTP, afim de comparar com o tempo despendido na comunicação serial do DCT. O experimento consiste em dois computadores em uma mesma rede local, conectados a um *switch* através de cabos. Um computador é o emissor e o outro computador é o receptor. O emissor envia uma mensagem e marca o tempo em que enviou, o receptor recebe a mensagem e envia uma resposta, o emissor recebe a resposta e marca o tempo

Figura 12 – Cenário do experimento Computador-Dispositivo.



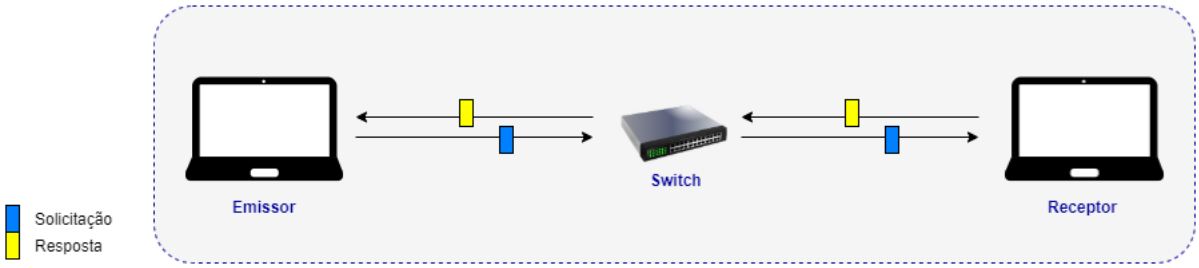

Fonte: Elaborado pelo Autor

em que recebeu. O tempo entre o envio e o recebimento é o tempo de ida e volta da mensagem. Novamente, a metade deste valor é o tempo que a mensagem levou para ser enviada entre o emissor e o receptor.

O emissor é o responsável pelas marcações de tempo no experimento, por isso foi alocação no computador com Sistema Operacional Linux, que entrega maior precisão de tempo se comparado com o Windows 10.

A Figura 13 ilustra o cenário do experimento Computador-Computador.

Figura 13 – Cenário do experimento Computador-Computador.




Fonte: Elaborado pelo Autor

No experimento foram executados 5 ensaios com 100 amostras cada, com intervalo de 5 segundos entre as amostras, totalizando 500 amostras. O experimento foi executado igualmente nos dois cenários.

O resultado dos experimentos podem ser vistos na Tabela 3. No cenário computador-dispositivo o tempo médio para o envio de mensagens foi de 31.33 microssegundos enquanto no cenário computador-computador o tempo médio para o envio de mensagens foi de 5.09 milissegundos.

Para o cenário Computador-Dispositivo a margem de erro foi de 0.32 $\mu$s, que permite estimar com confiança de 95% que a latência média encontra-se entre 31.01 $\mu$s e

| Cenário | Média | Desvio Padrão | Mínimo | Máximo |
|---|---|---|---|---|
| Computador-Dispositivo | 31.33 $\mu$s | 3.70 $\mu$s | 14.03 $\mu$s | 64.01 $\mu$s |
| Computador-Computador | 5.09 ms | 326 $\mu$s | 3.10 ms | 5.74 ms |

Tabela 3 – Resultado experimento de Latência.

31.61 $\mu$s. Para o cenário Computador-Computador a margem de erro foi de 28.58 $\mu$s, que permite estimar com confiança de 95% que a latência média encontra-se entre 5.06 ms e 5.12 ms.

Isso mostra que o tempo de atraso na marcação do tempo utilizando o DCT é em média mais de 100 vezes menor do que utilizando a solicitação de carimbo via HTTP, e esse número ainda pode ser muito maior em cenários com redes maiores.

## 7.3 TEMPO DE RESPOSTA

O experimento de tempo de resposta, diferente do experimento de latência, mediu o período desde a solicitação de carimbo do tempo pela aplicação até a obtenção da resposta com o certificado de carimbo do tempo assinado. O objetivo deste experimento é aferir a performance do dispositivo quanto a emissão de certificados de carimbo do tempo. O cenário é o mesmo da Figura 12 mas agora é medido também o tempo despendido pelo dispositivo para geração do certificado de carimbo do tempo e da assinatura.

O experimento consiste em solicitar a emissão de um carimbo do tempo e marcar o tempo que o dispositivo demorou para responder a solicitação. Foram executados 5 ensaios do experimento, em cada ensaio foram realizadas 100 solicitações, totalizando 500 amostras.

A Tabela 4 mostra o resultado do experimento. O tempo médio obtido foi de 542 ms.

| Média | Desvio Padrão | Mínimo | Máximo |
|---|---|---|---|
| 542.00 ms | 7.42 ms | 521 ms | 562 ms |

Tabela 4 – Resultado do experimento de tempo de resposta.

A margem de erro foi de 0.65 $\mu$s, que permite estimar com confiança de 95% que a precisão média encontra-se entre 541.35 ms e 542.65 ms. Esse resultado mostra que o DCT pode emitir até 110 carimbos por minuto.

## 7.4 PRECISÃO

O microcontrolador utilizado possui um RTC com precisão de microsegundos. Isso significa que o dispositivo pode fornecer tempos com até seis casas decimais, contudo deve-se levar em conta também o tempo que o microcontrolador necessita para realizar

uma marcação de tempo. A fim de descobrir esse tempo realizou-se um experimento onde o RTC é consultado ininterruptamente. Em cada consulta é anotado a diferença entre o valor retornado e o anterior. O experimento leva o microcontrolador a uma situação extrema com a finalidade de descobrir o máxima capacidade de marcação de tempo (granularidade de tempo), sendo esta considerada a precisão real do microcontrolador.

No experimento foram realizados 5 ensaios com 100 amostras cada, totalizando 500 amostras. A Tabela 5 mostra o resultado do experimento. O tempo médio o obtido foi de 16.91 $\mu$s.

| Média | Desvio Padrão | Mínimo | Máximo |
|---|---|---|---|
| 16.91 $\mu$s | 1.38 $\mu$s | 16 $\mu$s | 25 $\mu$s |

Tabela 5 – Resultado experimento de precisão do microcontrolador.

A margem de erro foi de 0.12 $\mu$s, que permite estimar com confiança de 95% que a precisão média encontra-se entre 16.79 $\mu$s e 17.03 $\mu$s.

## 7.5 SINCRONIZAÇÃO

A sincronização do RTC do dispositivo é realizado com um servidor de tempo (Vide seção 6.4) pelo método descrito na seção 5.3. Para aferir a eficiência da sincronização realizou-se um experimento que mediu a defasagem do relógio. O experimento consistiu em consultar o relógio do dispositivo logo após a sincronização. Um computador solicita via comunicação serial o tempo do relógio do dispositivo e marca a diferença de tempo entre o relógio do computador.

Foram executados 5 ensaios de 100 amostras cada, totalizando 500 amostras. A Tabela 6 mostra o resultado do experimento. O tempo médio obtido foi de 126.41 $\mu$s.

| Média | Desvio Padrão | Mínimo | Máximo |
|---|---|---|---|
| 126.41 $\mu$s | 725.49 $\mu$s | -6179 $\mu$s | 2508 $\mu$s |

Tabela 6 – Resultado do experimento de defasagem na sincronização.

A margem de erro foi de 63.59 $\mu$s, que permite estimar com confiança de 95% que a defasagem média após a sincronização encontra-se entre 62.82 $\mu$s e 190.00 $\mu$s. Apesar de o valor máximo do intervalo de confiança ser de décimos de milissegundos em alguns casos atingiu avanço ou atraso de milissegundos. Acredita-se que a defasagem é causada pelo método de sincronização que adota a simplificação de simetria do tempo gasto de ida e volta das mensagens. Além disso, o formato da rede onde se encontram o servidor de tempo e o dispositivo podem aumentar a defasagem. No experimento, o servidor de tempo e o dispositivo estavam na mesma rede local. Se o servidor de tempo estiver mais longe, os enlaces da rede podem causar esperas de mensagens em filas de roteadores e

switchs, possivelmente aumentando a diferença dos tempos de ida e volta das mensagens e a defasagem na sincronização.

## 7.6 ACURÁCIA

A acurácia de relógios é especificada por um valor em *ppm*(partes por milhão), significa que o relógio pode atrasar esse valor em uma unidade de tempo ao longo de 1 milhão da mesma unidade de tempo. Por exemplo, um relógio com acurácia de 10 ppm pode atrasar 10 milissegundos a cada aproximadamente 16.67 minutos (1 milhão de milissegundos).

O valor da acurácia implica na frequência necessária para sincronização do relógio. Para aferir a acurácia do dispositivo realizou-se um experimento. O experimento consistiu em fazer a sincronização do relógio como já mostrado anteriormente e acompanhar a defasagem por um período de tempo. Um computador consulta o relógio do dispositivo logo após a sincronização, aguarda 1 minuto, consulta novamente e anota a diferença dos tempos retornados.

Foram executados 5 ensaios de 100 amostras cada, totalizando 100 amostras. A Tabela 7 mostra o resultado do experimento. O tempo médio obtido foi de 2027.58 $\mu$s.

| Média | Desvio Padrão | Mínimo | Máximo |
|---|---|---|---|
| 2027.58 $\mu$s | 49.17 $\mu$s | 1853 $\mu$s | 2299 $\mu$s |

Tabela 7 – Resultado do experimento de acurácia.

A margem de erro foi de 9.64 $\mu$s, que permite estimar com confiança de 95% que a defasagem média ao longo de 1 minuto encontra-se entre 2017.94 $\mu$s e 2037.22 $\mu$s. Esse resultado implica em uma acurácia aproximada de 34 ppm.

# 8 CONSIDERAÇÕES FINAIS

O objetivo principal deste trabalho foi propor uma solução de carimbo do tempo alternativo ao modelo de carimbo do tempo das ICP-Brasil. Para atingir este objetivo dividiu-se o trabalho em algumas etapas. Inicialmente estudou-se o modelo de carimbo do tempo da ICP-Brasil. Nesta etapa foi possível ter uma visão geral do modelo de carimbo do tempo da ICP-Brasil e identificar problemas ou deficiências do modelo. Em seguida elaborou-se um o modelo alternativo visando corrigir ou melhorar as deficiências do modelo de carimbo de tempo da ICP-Brasil. A principal característica do modelo proposto é a utilização de um hardware compacto, de baixo custo e com criptografia pós-quântica. Após isso fez-se uma revisão bibliográfica para encontrar possíveis soluções relacionadas a proposta. Posteriormente realizou-se o desenvolvimento da proposta, com a implementação do firmware do dispositivo, da aplicação e do servidor de tempo. Na implementação dos softwares destaca-se a utilização do algoritmo assinatura digital XMSS. A implementação e a pesquisa pelo método de encapsulamento do microcontrolador fizeram parte de uma prova de conceito que visou mostrar a viabilidade de construção do DCT. Por fim, realizou-se alguns experimentos com o protótipo para avaliar características importantes como latência de comunicação, tempo de resposta, precisão de tempo, defasagem na sincronização e acurácia do relógio.

Em todas as etapas foi imprescindível os conhecimentos adquiridos no curso de Engenharia de Computação. Destaca-se a pesquisa, a elaboração de projetos, o desenvolvimento de software e a integração com hardware. Em contextos mais específicos destacam-se programação de computadores, programação de microcontroladores, sistemas embarcados, programação web, estruturas de dados, algoritmos, redes de computadores e estatística.

Com o estudo do modelo de carimbo do tempo da ICP-Brasil salientou-se os seguintes problemas: a centralização, devido a arquitetura do modelo e as poucas ACT disponíveis; e o método de obtenção do carimbo, que impede uma marcação de tempo mais precisa. A proposta elaborada visou solucionar os dois problemas adotando um dispositivo portátil ao usuário. Essa mudança permite capilarizar parte a infraestrutura de carimbo do tempo, amenizando os problemas da centralização. Também permite aumentar a precisão na marcação do tempo no carimbo devido ao método de obtenção local. Em contrapartida, a adoção do dispositivo impacta em vulnerabilidades na segurança do modelo. Uma possível solução para este problema foi o encapsulamento do microcontrolador com resina de resistência mecânica elevada. Outro item de segurança adotado foi o algoritmo de assinatura digital XMSS na emissão do certificado de carimbo do tempo e também no protocolo de sincronização do relógio do DCT.

Quanto aos experimentos realizados, permitiram conhecer melhor características importantes de sistemas de carimbo do tempo e também permitiram avaliar essas características no DCT desenvolvido. O principal resultado obtido neste trabalho é o atraso da

marcação de tempo pelo DCT ser em média mais de 100 vezes menor do que no modelo que utiliza comunicação de rede, como é o caso do modelo da ICP-Brasil. Esse resultado é importante pois considera-se a latência o fator limitante na precisão de tempo do carimbo. O atraso na marcação do tempo na ordem de centésimos de milissegundos possibilita carimbos do tempo com precisão de milissegundos ou até décimos de milissegundos. No experimento de precisão, a granularidade na marcação de tempo encontrada para o dispositivo é menor que o valor obtido para latência, portanto não é um fator limitante. Os resultados dos experimentos de acurácia e sincronização não foram bons quanto o esperado, contudo, para a acurácia é possível obter melhores resultados com a substituição do relógio do dispositivo e para a sincronização acredita-se ser possível em um trabalho futuro melhorar o método de sincronização com repetição de consultas ou redundância de servidores de tempo. O tempo de resposta não influencia diretamente a precisão do carimbo, apenas se a frequência de requisições for alta. Contudo, o número obtido de emissões por minuto é surpreendente grande para um microcontrolador de baixo custo. Parte disso deve-se a adoção do XMSS como algoritmo de assinatura digital.

Quanto a descentralização, o método de obtenção do carimbo localmente minimiza os problemas de indisponibilidade de serviço, escalabilidade e necessidade de infraestruturas avançadas por parte das ACT. Acredita-se também que um dispositivo com baixo custo é capaz de introduzir novos usuários e adoção em novas tecnologias.

Como trabalhos futuros, pretende-se comparar os SCT das ACT e os Módulos de Segurança Criptográfico por eles utilizados com o DCT desenvolvido através de experimentos. Além disso, buscar-se-ão alternativas para expandir a descentralização do sistema. Por exemplo, a utilização de servidores de tempo independentes, com sistema de homologação do dispositivo e processo de auditoria contínua

## 8.1 TRABALHOS FUTUROS

Como trabalhos futuros identificaram-se durante o desenvolvimento alguns itens que podem ser melhorados ou acrescentados para a evolução da proposta. São eles:

- **Número de Chaves Disponíveis.** A implementação permitiu armazenar apenas 512 de uma única vez. São ideias aumentar a memória disponível para armazenar mais chaves ou implementar um sistema seguro de renovação de chaves.

- **Maior descentralização.** A não dependência das ACT poderia trazer maior descentralização. Uma possibilidade é a utilização de servidores de tempo independentes e um sistema de homologação do dispositivo.

- **Aumentar a precisão de tempo.** A substituição do RTC ou a substituição do microcontrolador poderia aumentar a precisão de tempo.

- **Manufatura do invólucro.** Encapsulamento do microcontrolador com resina epóxi, blindagem térmica e blindagem eletromagnética. Etapa que envolve o estudo de técnicas de encapsulamento, principalmente referente às blindagens.

- **Processo de auditoria.** A realização de auditoria contínua poderia trazer maior segurança para a ACT no modelo proposto e também para a terceira parte. Também poderia trazer mais segurança para os participantes em um modelo de maior descentralização.

- **Método de sincronização.** Melhorar o método de sincronização, possivelmente alterando o protocolo para realizar múltiplas consultas e/ou redundância de servidores de tempo. Outra alternativa é estudar a possibilidade da utilização de outros protocolos, e.g. *Precision Time Protocol* (PTP) descrito na RFC 8173 (IETF, 2017), e a utilização de *Global Positioning System* (GPS) com o intuito de aumentar a acurácia na sincronização.

- **Experimento com SCT de uma ACT.** Consiste em repetir experimentos semelhantes aos já realizados neste trabalho, contudo com um SCT de uma ACT ou com infraestrutura equivalente. O objetivo é comparar os resultados do experimento com os resultados obtidos nos experimentos com o DCT.

- **Otimização do algoritmo WOTS.** Atualmente já existem variações mais otimizadas do algoritmo WOTS, como é mostrado por (PERIN *et al.*, 2018). Estas variações mais otimizadas podem permitir a redução do consumo de memória e também do tempo de resposta na realização da assinatura.

- **Aprimorar a infraestrutura de testes.** Utilização de sistemas de tempo real ou driver com prioridade no escalonador de processos do sistema operacional, com o objetivo de realizar medidas mais confiáveis. Outra possibilidade é a utilização de um analisador lógico, que permitirá mapear o consumo de tempo de cada operação dentro do microcontrolador.

- **Atualização das linguagens.** Eliminar o uso das linguagens de programação interpretadas na aplicação e no servidor de tempo, substituindo por linguagens compiladas, e.g. C/C++. O objetivo é eliminar possíveis perdas de tempo causadas pelo *garbage collector* dessas linguagens e dessa forma melhorar a acurácia da sincronização e diminuir o atraso na solicitação do carimbo do tempo.

# REFERÊNCIAS

BROBY, Daniel; BASU, Devraj; ARULSELVAN, Ashwin. The Role of Precision Timing in Stock Market Price Discovery when Trading through Distributed Ledgers. **Journal of Business Thought**, v. 10, 2019. DOI: 10.18311/jbt/2019/23355.

BUCHMANN, A. Johannes; KARATSIOLIS, Evangelos; WIESMAIER, Alexander. **Introduction to Public Key Infrastructures**. 1. ed. New York - Dordrecht - London: Springer, 2013.

COOPER, David A. *et al.* Recommendation for Stateful Hash-Based Signature Schemes. **Draft NIST Special Publication 800-208**, 2019. DOI: 10.6028/NIST.SP.800-208-draft.

COULOURIS, George *et al.* **Sistemas Distribuídos - Conceitos e Projetos**. 5. ed. Porto Alegre: Bookman, 2013.

CRYPTO, ID. **Blockchain não é apenas bitcoin! A tecnologia pode ajudar seu negócio!** 2019. Disponível em: https://cryptoid.com.br/blockchain/blockchain-nao-e-apenas-bitcoin-a-tecnologia-pode-ajudar-seu-negocio/. Acesso em: 23 jan. 2020.

_____. **Governo Federal homologa primeiro sistema de Carimbo do Tempo brasileiro**. 2016. Disponível em: https://cryptoid.com.br/banco-de-noticias/governo-federal-homologa-primeiro-sistema-de-carimbo-do-tempo-brasileiro/. Acesso em: 21 jan. 2020.

_____. **IoT começa a demandar certificação digital**. 2020. Disponível em: https://cryptoid.com.br/identidade-digital-destaques/iot-comeca-a-demandar-certificacao-digital/. Acesso em: 23 jan. 2020.

ESPRESSIF SYSTEMS. **ESP32 Series**: Datasheet. [*S.l.*], jan. 2020.

EUROPEAN TELECOMMUNICATION STANDARD INSTITUTE. **TS 101 861**: Electronic Signatures and Infrastructures (ESI); Time stamping profile. [*S.l.*], jul. 2011.

FAN, Lianhua; WONG, C. P. Thermosetting and thermoplastic bisphenol A epoxy/phenoxy resin as encapsulant material. **Proceedings International Symposium on Advanced Packaging Materials Processes, Properties and Interfaces**, 2001. DOI: 10.1109/ISAOM.2001.916580.

GIPP, Bela; MEUSCHKE, Norman; GERNANDT, André. Decentralized Trusted Timestamping using the Crypto Currency Bitcoin., 2015. Disponível em: https://arxiv.org/abs/1502.04015. Acesso em: 02 de julho de 2020.

HADIZADEH, Rameen *et al.* Embedded Component Packaging for Wafer-Level Encapsulated and Integrated RF Mems. **2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems & Eurosensors XXXIII**, 2019. DOI: 10.1109/TRANSDUCERS.2019.8808808.

HARMANN, Peter. Distributed Trusted Timestamp. **Master's Thesis, Masaryk University**, 2019. Disponível em: https://is.muni.cz/th/bgmmw/Trusted_Timestamping_v3_is.pdf. Acesso em: 02 de julho de 2020.

INFRAESTRUTURA DE CHAVES PÚBLICAS BRASILEIRA. **Manual de Condutas Técnicas 3 – Volume I**: Requisitos, Materiais e Documentos Técnicos para Homologação de Tokens Criptográficos no Âmbito da ICP-Brasil. [*S.l.*], set. 2017.

_____. **PROCEDIMENTOS PARA AUDITORIA DO TEMPO NA ICP-BRASIL**: DOC-ICP-14 versão 1.3. [*S.l.*], set. 2015.

_____. **REQUISITOS MÍNIMOS PARA AS DECLARAÇÕES DE PRÁTICAS DAS AUTORIDADES DE CARIMBO DO TEMPO DA ICP-BRASIL**: DOC-ICP-12 versão 1.3. [*S.l.*], dez. 2019.

_____. **REQUISITOS MÍNIMOS PARA AS POLÍTICAS DE CARIMBO DO TEMPO DA ICP-BRASIL**: DOC-ICP-13 versão 1.2. [*S.l.*], dez. 2015.

_____. **VISÃO GERAL DO SISTEMA DE CARIMBOS DO TEMPO NA ICP-BRASIL**: DOC-ICP-11 versão 1.3. [*S.l.*], set. 2015.

INTERNET ENGINEERING TASK FORCE. **Request for Comments: 8173**: Precision Time Protocol Version 2 (PTPv2). [*S.l.*], jun. 2017.

INTERNET RESEARCH TASK FORCE. **Request for Comments: 8391**: XMSS: eXtended Merkle Signature Scheme. [*S.l.*], jan. 2020.

ITI, Instituto Nacional de Tecnologia da Informação. **Autoridades de Carimbo do Tempo**. 2017. Disponível em: https://www.iti.gov.br/icp-brasil/autoridades-de-carimbo-do-tempo. Acesso em: 21 jan. 2020.

_____. **Carimbo do Tempo**. 2017. Disponível em: https://www.iti.gov.br/acesso-a-informacao/41-perguntas-frequentes/131-carimbo-do-tempo. Acesso em: 20 jan. 2020.

_____. **Certificado Digital - Saiba Mais**. 2020. Disponível em: https://aquitemcd.iti.gov.br/certificado-digital/. Acesso em: 29 jan. 2020.

_____. **Equipamentos Certificados**. 2017. Disponível em: https://www.iti.gov.br/homologacao/64-homologacao/212-equipamentos-homologados. Acesso em: 21 jan. 2020.

_____. **Uso do carimbo do tempo é oficial no Brasil**. 2008. Disponível em: https://www.iti.gov.br/noticias/68-iti-na-midia/1602-uso-do-carimbo-do-tempo-e-oficial-no-brasil. Acesso em: 20 jan. 2020.

KAKEI, Shohei *et al.* Offline Time-Stamping System: Its Design and Implementation. **2012 IEEE International Conference on Control System, Computing and Engineering**, 2012. DOI: 10.1109/ICCSCE.2012.6487179.

KUROSE, Jim; ROSS, Keith. **Computer Networking - A Top-Down Approach**. 6. ed. United States of America: Pearson, 2013.

LI, Qing; YAO, Caroline. **Real-Time Concepts for Embedded Systems**. 1. ed. San Francisco - USA: CMP, 2003.

MDN, Contributors. **An overview of HTTP**. 2020. Disponível em: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview. Acesso em: 27 fev. 2020.

MENEZES, Alfred J.; OORSCHOT, Paul C. van; VANSTONE, Scott A. **Handbook of Applied Cryptography**. 1. ed. [*S.l.*]: CRC Press, 1996.

NATIONAL INSTITUTE OF STANDARDS e TECHNOLOGY. **SECURE HASH STANDARD**: Announcing the. [*S.l.*], ago. 2002.

NETWORK WORKING GROUP. **Request for Comments: 3161**: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). [*S.l.*], ago. 2001.

_____. **Request for Comments: 3628**: Policy Requirements for Time-Stamping Authorities (TSA). [*S.l.*], nov. 2003.

NEUMANN, Christoph; HEEN, Oliver; ONNO, Stéphane. DNStamp: Short-lived trusted timestamping. **Computer Networks. 2014, vol. 64**, 2014. DOI: 10.1016/j.comnet.2014.02.016.

NIC.BR. **O NTP**. 2020. Disponível em: https://ntp.br/ntp.php. Acesso em: 20 fev. 2020.

NSC, Total. **UFSC é a primeira universidade brasileira a emitir diploma digital lançado pelo MEC**. 2020. Disponível em: https://www.nsctotal.com.br/noticias/ufsc-e-a-primeira-universidade-brasileira-a-emitir-diploma-digital-lancado-pelo-mec. Acesso em: 20 jan. 2020.

OKUNO, A.; FUJITA, N.; ISHIKANA, Y. Low cost and high reliability extremity CSP packaging technology. **1999 Proceedings. 49th Electronic Components and Technology Conference**, 1999. DOI: 10.1109/ECTC.1999.776363.

OLIVEIRA, Gabriel Estevam; FRANCO, Alvaro Junio Pereira. Uma Apresentação Sucinta do Sistema de Criptografia RSA. **6º Simpósio de Integração Científica e Tecnológica do Sul Catarinense – SICT-Sul**, 2017.

OPENJS, Fundation. **NodeJS**. 2020. Disponível em: https://nodejs.org/en/. Acesso em: 17 mar. 2020.

PERIN, Lucas P. *et al.* Tuning the Winternitz hash-based digital signature scheme. **2018 IEEE Symposium on Computers and Communications (ISCC)**, 2018. DOI: 10.1109/ISCC.2018.8538642.

PYTHON SOFTWARE FOUNDATION, Community. **Python**. 2020. Disponível em: https://www.python.org/about/. Acesso em: 17 mar. 2020.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. **Communications of the ACM**, 1978. DOI: 10.1145/359340.359342.

SANTIAGO, Christopher. **Soluti Responde - ACT: o que é e como aplicar um carimbo do tempo?** 2019. Disponível em: https://solutiresponde.com.br/act-o-que-e-e-como-aplicar-um-carimbo-do-tempo/. Acesso em: 20 jan. 2020.

SCHILDT, Herbert. **C, completo e total**. 3. ed. São Paulo: Makron Books, 1996.

SHOSTACK, Adam. **Threat Modeling: Designing for Security**. 1. ed. Indianapolis - USA: Wiley, 2014.

STARNBERGER, Guenther; FROIHOFER, Lorenz; GOESCHKA, Karl M. Using Smart Cards for Tamper-Proof Timestamps on Untrusted Clients. **2010 International Conference on Availability, Reliability and Security**, 2010. DOI: 10.1109/ARES.2010.78.

TAO, Zhiqiang *et al.* Synthesis and Characterization of Novel Multiaromatic Epoxy Resin for Advanced Microelectronic Packaging Applications. **2007 8th International Conference on Electronic Packaging Technology**, 2007. DOI: 10.1109/ICEPT.2007.4441455.

TRIOLA, Mario F. **Introdução à estatística: atualização da tecnologia**. 11. ed. Rio de Janeiro: LTC, 2013.

VIGIL, Martín *et al.* Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: A survey. **Computers & Security**, v. 50, p. 16–32, 2015. DOI: 10.1016/j.cose.2014.12.004.

YAMOAKA, S.; KUSUHARA, A.; OKABE, Y. Liquid dropping resin for IC encapsulation. **IEEE Transactions on Components, Hybrids, and Manufacturing Technology**, 1988. DOI: 10.1109/33.2978.