Provendo controle de acesso para o Sistema de Telessaúde Holográfico da UFF através de um protocolo de autenticação de três fatores

Nicolas F. M. da Silva[1] **e Natalia C. Fernandes**[1]

[1]Escola de Engenharia – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brasil

{nicolasfulli,nataliacf}@id.uff.br

Abstract. *The Holographic Telehealth System of the Universidade Federal Fluminense (STH-UFF) aims to offer medical support to places with a low contingency of specialties through tele-interconsultations. The system has been used in Brazilian Army patients in controlled environments and should be expanded to public health units. However, this expansion increases the risk of the consultations' confidential data exposure. In this article, an architecture and implementation of an access control system for STH-UFF is proposed, based on a multi-factor authentication. We analyze security requirements, project's costs, the computational feasibility and trade-offs of the developed system.*

Resumo. *O Sistema de Telessaúde Holográfico da Universidade Federal Fluminense (STH-UFF) tem como objetivo oferecer apoio médico para locais com baixo contingente de especialidades através da realização de teleinterconsultas. O sistema vem sendo utilizado em ambientes controlados para pacientes do Exército Brasileiro e deve ser ampliado para unidades públicas de saúde. Contudo, essa ampliação acarreta no aumento da ameaça à exposição dos dados confidenciais das consultas. Neste artigo, é proposta uma arquitetura e implementação de um sistema de controle de acesso para o STH-UFF, baseado em uma autenticação multifator. São analisados os requisitos de segurança, o custo do projeto, a viabilidade computacional da aplicação e os trade-offs do sistema desenvolvido.*

1. Introdução

O Sistema de Telessaúde Holográfico (STH) é um projeto desenvolvido pela Universidade Federal Fluminense com o objetivo de promover atendimento médico especializado para locais remotos do território brasileiro através de uma técnica de projeção holográfica que proporciona ao médico consultor uma sensação tridimensional da imagem do paciente, aumentando sua capacidade analítica para prover diagnósticos ou orientar um profissional no ambiente de atendimento remoto [Bello 2016].

O sistema opera com um servidor de controle gerenciando dois tipos de consultórios, o Consultório de Saúde Virtual (CSV) e o Consultório Holográfico Virtual (CSH), onde se situam respectivamente paciente junto ao médico local e a bancada de médicos especialistas. Toda gerência e controle de mídias e rede é implementada pela ferramenta STH-UFF [Fonseca 2019], que promove a segurança dos dados sensíveis dos pacientes consultados através de uma VPN *site-to-site* entre CSV e CSH. Sua utilização

atual é restrita a ambientes controlados, oferecendo suporte ao Hospital Central do Exército (HCE) e atividades acadêmicas realizadas em conjunto com o Hospital Universitário Antônio Pedro (HUAP). Em 2017, o projeto sofreu mudanças em sua arquitetura com objetivo de transformá-lo em um projeto de baixo custo para que pudesse ser economicamente viável ampliá-lo para atender um número maior de localidades, além de ajustes para permitir que o mesmo fosse capaz de operar com recursos de banda extremamente limitados [Beaklini and Fonseca et. al 2018]. Com tais mudanças, complementares ao sucesso das ações supracitadas, estuda-se sua ampliação para auxílio na rede pública de saúde do município de Niterói.

A segurança de dados é uma das principais preocupações no ramo dos serviços de teleinterconsultas. Apesar da aplicação de VPNs *site-to-site* fornecer a integridade e disponibilidade das informações que trafegam entre os sistemas devido ao tunelamento do enlace, somente tal medida não torna o sistema capaz de atender a todos os fundamentos necessários para promover segurança a um sistema de informação [NBR27001 2006] [Stallings 2015]. O fornecimento de autenticidade e confiabilidade ao canal de comunicação é vulnerável a ataques devido aos *endpoints* não estarem protegidos por nenhum mecanismo de controle de acesso. Além disso, fora de um ambiente controlado, as informações enviadas dentro do sistema estão expostas a inúmeras ameaças como personificação de servidor, exposição do usuário, entre outros [He et al. 2015] que podem explorar brechas de algum mecanismo de controle de acesso que possa vir a ser implementado, tornando ideal que o mesmo seja elaborado para possuir resistência contra a maior parte dos ataques conhecidos, sempre tentando ao máximo manter as características de baixo custo e baixo consumo de banda do sistema.

O controle de acesso atual da ferramenta STH-UFF é baseado somente em um fator e pode ser facilmente quebrado através de ataques cibernéticos clássicos, como ataque de dicionário e ataque de usuário privilegiado. Assim, para prover as necessidades de segurança do sistema é proposta uma aplicação de autenticação de três fatores baseada no trabalho de [Jiang et al. 2018], chamada de STH-HELENA, que além de permitir uma integração com a ferramenta de controle atual de sistema, é capaz de preservar o caráter de baixo custo do projeto STH-UFF.

A implementação da ferramenta deve possibilitar a manutenção de características chave, que são:

1. *Resistência a ataques de perda de cartão*: Usuários não autorizados não são capazes de, ao portar um cartão de uma vítima, utilizá-lo para obter acesso a sua senha ou ao sistema.
2. *Autenticação Mútua*: O usuário e o servidor são capazes de autenticar um ao outro.
3. *Anonimato do Usuário*: O sistema deve promover a proteção da identidade do usuário e prevenir seu rastreamento.

Neste artigo são apresentados os recursos escolhidos e adotados para implementação de cada etapa do protocolo na ferramenta desenvolvida, com o detalhamento de suas escolhas e trade-offs adotados para permanência dentro das premissas definidas para o projeto e das características chave para um sistema de autenticação seguro. Além disso foram realizados testes para avaliação do impacto dos *trade-offs* nos aspectos de segurança e eficiência do sistema, como também a viabilidade computacional da ferramenta proposta.

O restante do artigo está organizado como descrito a seguir. Na Seção 2, são apresentados os principais trabalhos relacionados da literatura. Na Seção 3, é apresentada a arquitetura da solução proposta. Na Seção 4 detalham-se os trade-offs e a operação da ferramenta, enquanto que na Seção 5, são descritos os resultados de sua avaliação. Por fim, a Seção 6 conclui o artigo.

2. Trabalhos Relacionados

O primeiro modelo de autenticação foi proposto por [Lamport L 1981], possuindo senha como seu único fator. A partir dele, muitos outros protocolos foram desenvolvidos utilizando pelo menos um ou uma combinação de três categorias de fatores de autenticação: 1) algo que o usuário sabe - uma senha; 2) algo que o usuário tem - um *smart card*; e 3) algo que o usuário é - uma característica biométrica. Com a premissa de que um possível invasor não possuirá previamente pelo menos um destes fatores.

Conforme progrediu-se o desenvolvimento dos modelos, muitos protocolos de um e dois fatores mostraram-se vulneráveis a ataques de personificação e principalmente ataques de dicionário e força bruta quando ofereciam alteração *offline* de senhas. Foi então que [Wang 2016] propôs um protocolo de autenticação de dois fatores baseado em senha e *smart-card* capaz de suportar ataques *offline* através do desenvolvimento de um ”verificador difuso”.

Inicialmente, em sistemas de telessaúde, os protocolos utilizados eram de dois fatores baseados em senha e *smart-card*, nos quais a validação simultânea de ambos os fatores era necessária para confirmação da identidade do usuário. Contudo, cartões físicos podem ser perdidos e os parâmetros de autenticação armazenados nos mesmos podem ser duplicados [Jiang et al. 2018]. Como solução para tal problema, protocolos envolvendo três fatores foram propostos [Tan Z 2013], nos quais a validação simultânea dos três fatores se faz necessária.

Com a crescente intensificação de estudos no ramo da autenticação de três fatores, surgiu uma cadeia de propostas de modelos seguros a serem implementados nos sistemas de telessaúde. Dois dentre estes modelos se destacam por proporcionarem um baixo custo computacional de execução [Arshad H and Nikooghadam M 2014][Lu et al. Y 2014]. Contudo, ambos possuem a mesma falha apresentada pela grande maioria dos protocolos de três fatores, que é a vulnerabilidade a ataques *offline*.

Tomando como base essa falha comum, [Jiang et al. 2018] propõe uma versão aprimorada do modelo de Lu, utilizando do ”verificador difuso” para superar sua principal vulnerabilidade. Na etapa de troca de chaves, o modelo herda de seus predecessores a cifra de chaves assimétricas por curvas elípticas (*ECC*), proporcionando chaves de tamanho muito reduzido comparado a chaves RSA de mesmo nível de segurança.

A proposta desse trabalho é uma aplicação da autenticação de três fatores dentro do contexto de aplicações de computação para a saúde. Nesse sentido, buscou-se um protocolo resistente aos principais ataques, apresentado na proposta de [Jiang et al. 2018], adaptando-se a proposta ao contexto das teleinterconsultas.

3. STH-HELENA - Sistema de Autenticação Multifator para Teleinterconsultas

O sistema proposto, o STH-HELENA, baseia-se na adaptação da proposta de [Jiang et al. 2018] para a autenticação de médicos e pacientes em teleinterconsultas. Nesse contexto, uma falha de autenticação pode levar ao vazamento de informações sigilosas do paciente, além de fraudes no protocolo médico. Assim, buscou-se encontrar uma solução segura e de baixo custo, para viabilizar a implantação da solução em larga escala em regiões carentes do interior do Brasil, nas quais o Sistema de Telessaúde Holográfico vem sendo aplicado.

3.1. Aplicação do Protocolo de Jiang et al. em Teleinterconsultas

O objetivo desta seção é detalhar as etapas envolvidas no processo de autenticação de [Jiang et al. 2018]. O protocolo adotado como base do processo de autenticação divide-se em três etapas: 1) Registro; 2) *Login*; e 3) Troca de Senha.

A notação utilizada no protocolo adaptado encontra-se descrita na Tabela 1. Para a inicialização do sistema, o servidor S gera uma chave privada principal x e um inteiro $2^4 \leq l \leq 2^8$ como parâmetro do verificador difuso, e então computa xP como sua chave pública.

Tabela 1. Notação utilizada para descrever o protocolo de autenticação em três fatores.

$\mathbf{U}_i$	*Médico usuário acessando o sistema a partir do CSV ou CSH*
$\mathbf{ID}_i$	*Identidade do médico usuário*
$\mathbf{PW}_i$	*Senha definida pelo médico usuário*
$\mathbf{B}_i$	*Biometria do médico usuário*
S	*Servidor de gerência do STH-UFF*
x	*Chave privada do servidor*
BH()	*Função Biohash*
$\mathbf{h}_1(), \mathbf{h}_2()$	*Funções hash utilizadas*
$\oplus$	*Operação OU Exclusivo bit a bit*
$\|\|$	*Operação de concatenação*

3.1.1. Registro

Na fase de registro do protocolo, primeiramente, o médico usuário U_i seleciona um nome identificador ID_i e envia ID_i para o servidor S. Após receber o requerimento de registro, S gera um número aleatório n_i e computa $K_i = h_i(ID_i||x||n_i)$.

Na sequência, S produz um *smart card* SCi contendo $Ki, xP, l, h_1(.), h_2(.), BH(.)$ para U_i. S mantém uma base de dados armazenando cada parâmetros ID_i, n_i dos $U_i, \forall i \in N$, onde N é o conjunto de todos os usuários do sistema.

Para se autenticar, o usuário U_i introduz o cartão em um leitor de cartões. Então, ele insere o ID_i, a senha PW_i e coleta a biometria Bi. O sistema do cartão, então, computa $V_i = h_2(h_2(ID||PW_i||BH(B_i))\ mod\ l)$ e $K_i' = K_i \oplus h_1(ID_i||PW_i||BH(B_i))$

Por fim, o cartão armazena $V_i, K_i', xP, l, h_1(.), h_2(.), BH(.)$ em sua memória e apaga K_i.

3.1.2. *Login*

Após o registro, o usuário pode se autenticar no sistema. Para tanto, primeiramente, o servidor S registra um número aleatório como chave privada d_s, depois S computa e envia a chave pública d_sP para U_i. Na sequência, U_i conecta o seu *smart card* SC_i em um leitor de cartão, insere sua identidade ID_i, sua senha PW_i e coleta sua biometria B_i. O sistema do *smart card* computa $X = h_2(h_2(ID_i||PW_i||BH(B_i))\ mod\ l)$ e verifica se $X = V_i$.

Caso sim, SC_i seleciona um número aleatório d_u, computa $K_i = K_i' \oplus h_1(ID_i||PW_i||BH(B_i))$, $AID_i = ID_i \oplus h_1(d_uP||d_ud_sP)$, $SK = h_1(ID_i||K_i||d_ud_sP)$, $M_1 = d_uxP$ e $M_2 = h_1(ID_i||d_uP||d_sP||K_i)$. Transmitindo (AID_i, M_1, M_2) para S.

Ao receber essa mensagem, S calcula $d_uP = x^{-1}M_1$ e deriva ID_i por meio de $ID_i = AID_i \oplus h_1(d_uP||d_ud_sP)$. Posteriormente, S computa $K_i = h_1(ID_i||x||n_i)$ e compara se $h_1(ID_i||d_uP||d_sP||K_i)$ é igual a M_2. Caso seja igual, S computa $SK = h(ID_i||K_i||d_ud_sP), M_3 = h_1(K_i||d_uP||SK)$. Por fim, S envia (M_3) para U_i.

Com o recebimento da mensagem, U_i verifica se $h_1(K_i||d_uP||SK)$ é igual a M_3 na mensagem recebida. Caso sim, U_i e S são mutuamente autenticados.

3.1.3. Troca de Senha

Caso um usuário legal queira mudar sua senha, U_i insere seu *smart card* no dispositivo e submete ID_i, PW_i e B_i. O *smart card* verifica se $V_i? = h_2(h_2(ID_i||PW_i||BH(B_i))\ mod\ l)$. Caso sim, U_i seleciona uma nova senha PW_i^{new}, SC_i, computa $V_i^{new} = h_2(h_2(ID_i||PW_i||BH(B_i))\ mod\ l)$ e $K_i'^{new} = K_i' \oplus h_1(ID_i||PW_i||BH(B_i) \oplus h_1(ID_i||PW_i^{new}||BH(B_i)$, e, por fim, atualiza V_i com V_i^{new} e K_i com K_i^{new}.

3.1.4. Verificador Difuso

O verificador difuso, foi proposto por [Wang 2016], e é um método para evitar ataques *offline* gerados pela perda do *smart card* do usuário. Seu funcionamento se baseia na combinação de um segredo $K_i' = K_i \oplus h_1(ID_i||PW_i)$ com a computação de um parâmetro de verificação pela forma $V_i = h_2(h_2(ID_i||PW_i)\ mod\ l)$, com o valor de l entre $2^4 \leq l \leq 2^8$. Utilizando esta forma, pode ser assegurado que existem $\frac{|D_{id}|*|D|}{l} \approx 2^{32}$ candidatos ao par (ID, PW) para gerar dificuldade a algum adversário A quando $|D_{id}| = |D| = 10^6$ e $l = 2^8$, no qual $|D_{id}|$ e $|D|$ denotam o tamanho do espaço de identidade e do espaço de senha, respectivamente. Dessa maneira, ainda que a identidade do usuário já tenha sido descoberta, A ainda será frustrado por existirem $\frac{|D|}{n_o} \approx 2^{12}$ candidatos a senha. Para, então, excluir as senhas irreais dentre as 2^{12} opções restantes, não existe outra saída a não ser lançar um ataque *online* para descoberta da senha interagindo com S para determinar a correta. Caso A decida por continuar com o ataque online, o mesmo tem um chance extremamente alta de optar por uma das senhas irreais, o que faria com

que seu segredo K_i' não fosse computado corretamente, frustrando A e permitindo com que o servidor detecte possíveis violações de cartão, podendo tomar medidas de segurança baseadas nestas detecções.

3.2. Equipamentos e Módulos Utilizados

A arquitetura atual dos consultórios remotos, chamados de CSV [Beaklini and Fonseca et. al 2018] é composta de:

- Computador de sistema operacional Linux e distribuição Ubuntu;
- Ponto de acesso TP-Link TL-WR842ND com o *framewok* OpenWRT instalado;
- Webcam Logitech HD Pro C920 como dispositivo de captura de vídeo em *full* HD;
- Microfones e caixas sem grandes requisitos técnicos;

Durante a implementação da aplicação de autenticação, procurou-se aproveitar ao máximo os recursos atualmente existentes na arquitetura dos CSVs já que, por comporem um contingente maior de instalações, qualquer acréscimo em sua implementação poderia prejudicar a característica de escalabilidade do projeto. Com essa premissa tornou-se necessário desenvolver dois métodos alternativos para coleta biométrica e uso de smart cards.

Como alternativa para a leitura biométrica através de digitais, avaliou-se que era possível a utilização de método de coleta biométrica facial ou por voz, utilizando respectivamente a câmera e o microfone já presentes na arquitetura do CSV. Dentre as duas soluções encontradas, os fatores mais influentes na precisão da identificação dos usuários foram luminosidade no caso da biometria facial, e ruído externo no caso da biometria por voz. Como o controle da luminosidade dos CSV pode ser solucionada de forma mais simples que o controle do ruído, optou-se pela implementação da biometria facial.

Para poder suprir a utilização dos *smart cards*, a alternativa encontrada precisaria ser algo que o usuário fosse capaz de acessar com facilidade e que armazenasse de forma segura todas as informações necessárias para a autenticação. Posto isso, analisou-se a utilização de Códigos QR (*Quick Response*) devido a sua capacidade de armazenamento de informação.

A ferramenta de gerência que funciona como sistema base para a teleinterconsulta do STH, descrita em [Fonseca 2019], foi construída utilizando a linguagem de programação Python 3. Portanto, a solução para o *framework* de controle de acesso também foi construída nesta linguagem.

Dentro da solução, alguns módulos principais foram utilizados:

- *PyCrypto* - Utilizada através do PyCryptodome. Um pacote de Python independente, com primitivas criptográficas de baixo nível, que permite, através de sua API, a utilização de várias funcionalidades de criptografia, subdividas em pacotes dedicados para auxiliar cada classe específica de soluções. Dentro das necessidades do protocolo adotado, foram utilizados cinco subpacotes:
    1. **Hash**: Este subpacote tem como objetivo fornecer as mais modernas funções de hash. SHA3 e Blake são famílias modernas de *hash*, sendo suas versões de SHA3-256 e BLAKE2 as de menor tamanho de *digest* consideradas seguras. Cabe destacar que, na proposta de [Jiang et al. 2018],

são utilizadas três funções *hash* independentes e indefinidas $H_1()$,$H_2()$ e $BH()$. Para defini-las, foi necessária a análise de suas funcionalidades no protocolo proposto. Como a função $H_1()$ é utilizada em todos os processos do servidor, e o mesmo está sujeito ao recebimento de muitas solicitações de autenticação simultâneas, optou-se pela função BLAKE2b em sua implementação. A função $H_2()$ é utilizada apenas pelo sistema do médico usuário e apenas em um processo que é sensível a ataques *offline*, o que torna interessante a opção por uma função hash que possua uma velocidade de execução mais lenta para retardar o desempenho de um possível adversário. Portanto, optou-se pela função SHA3-256. Os mesmos *hash* e raciocínio foram utilizados para a função $BH()$.

2. **PublicKey**: Subpacote para utilização de sistemas de criptografia de chaves assimétricas. Nesse pacote é implementado o sistema de *ECC*, que é uma família moderna de sistemas de criptografia de chaves assimétricas baseadas nas propriedades algébricas das curvas elípticas sobre campos finitos e na dificuldade de solução do Problema de Logaritmo Discreto para Curvas Elípticas (ECLDP) [Galvão 2006]. O *ECC* é capaz de implementar as principais funcionalidades de criptografias assimétricas e é considerado como um sucessor natural do sistema RSA, por ser capaz de fornecer chaves de tamanho inferior com a garantia do mesmo nível de segurança, proporcionando também geração e troca de chaves mais velozes.
3. **Cipher**: Subpacote que permite a utilização de algoritmos de encriptação simétricos, assimétricos e híbridos para proteção de dados sensíveis.

- *QR Code* - No sistema implementado, os códigos QR são gerados utilizando o pacote de python PyQRcode. Esse pacote permite que, após criados, os códigos QR possam ser salvos como SVG, PNG (por meio do módulo pypng) e possam ser exibidos diretamente na maioria dos terminais de distribuição Linux. Já sua leitura é realizada através do pacote Pyzbar, também capaz de ler códigos de barra.
- *Biometric* - É um algoritmo desenvolvido por [Geitgey 2018] utilizando Python, OpenCV e *deep learning*. O algoritmo realiza a codificação das faces através de uma rede neural treinada previamente com base em um *dataset* de mais de três milhões de imagens. A rede produz como saída de cada imagem um vetor de características de 128 valores reais utilizados para quantificar cada face encontrada e, posteriormente, utiliza a metodologia de *deep metric learning* para realizar a classificação das faces. Para iniciar a codificação, o algoritmo realiza a detecção das faces em imagens localizadas nos *datasets* de interesse. Essas detecções são realizadas através do método de Histograma de Gradientes Orientados(HOG), escolhido por possuir um custo de memória inferior ao método de Redes Neurais Convolucionais (CNN), o que a princípio o torna mais adequado aos requisitos do projeto. Em sequência, o algoritmo é capaz de realizar a identificação de faces em tempo real ao capturar um vídeo com a face do usuário, no qual os *frames* são coletados e o método de detecção de faces é aplicado. O método utilizado nesta etapa é o Haar Cascades que, apesar de antiquado, requer um custo de memória bastante inferior que outros métodos para detecções em tempo real e, consequentemente, viabiliza a utilização de *hardwares* mais acessíveis financeiramente. O principal ponto negativo do método é a baixa precisão para detecção de faces anguladas, porém tal situação pode ser facilmente controlada pelo médico usuário.

Além destes citados a ferramenta também utiliza os módulos *Email.MIME* para envio de mensagens por e-mail e *MySQL* para armazenamento dos $ID's$ e seus respectivos $ni's$. A Figura 1 exibe uma visão geral da ferramenta desenvolvida.

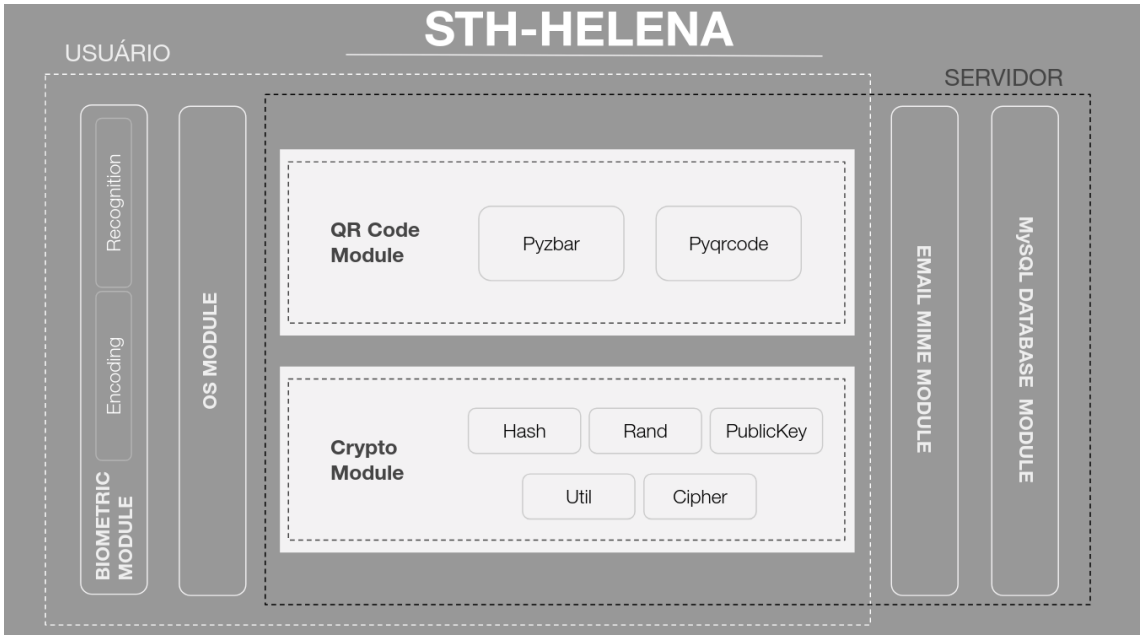


Figura 1. Arquitetura da ferramenta de autenticação desenvolvida.

4. Análise dos *Trade-Offs* na Definição dos Parâmetros do Sistema

Visando manter o caráter de baixo custo do projeto STH-UFF, duas grandes mudanças foram adotadas. Para sua realização, foram necessárias novas soluções para buscar manter as características de segurança do sistema.

4.1. *Smart Card* $\rightarrow$ *Código QR*

Uma das premissas do projeto STH-UFF é uma arquitetura de baixo custo que possa ser acessível para implementação em áreas com situações financeiras por vezes insatisfatórias. Visto tal proposição, os *smart card*, que possuem alto custo de implementação e manutenção, foram substituídos por códigos QR.

A solução obtida por meio da utilização de QR *Codes* parte da premissa que os médicos usuários possuem um celular com câmera. Visto que, segundo dados do IBGE, o Brasil possui cerca de dois celulares por cidadão e que dentre pessoas com mais de 10 anos, 77,1% possuem pelo menos um *smartphone* próprio [IBGE 2020], a premissa foi considerada viável por englobar a maioria da população.

Com a ferramenta proposta, após a etapa de registro, a emissão de *smart cards* para os médicos usuários é então substituída pelo envio, por meio eletrônico, dos QR *Codes*. Posteriormente, os médicos poderão exibir os códigos através de *smartphones*, que seriam capturados pelas câmeras dos consultórios e processados.

Apesar da natureza prática e conjunto de características úteis e hábeis para o sistema de telessaúde, os QR *Codes* apresentam a limitação de, uma vez gerados, não poderem ter suas informações alteradas. Tal restrição impacta diretamente nas etapas de

ativação do cartão e troca de senha do protocolo alterado pois ambas estão interligadas ao gerenciamento de informações previamente implantadas.

Para contornar a situação, foi avaliado o processo de reemissão e reenvio do código QR. Porém, o procedimento não mostrou-se capaz de atender integralmente as necessidades do protocolo adotado pois o critério de troca de senha *offline* não é atendido.

A solução final proposta utiliza-se da premissa supracitada, com o acréscimo do seguinte processo: durante as etapas de ativação e troca de senha, é gerado um novo código QR, que será exposto na tela do sistema e deverá ser fotografado e armazenado pelos médicos usuários.

4.1.1. Operação da Ferramenta com QR *Code*

Definidos os *trade-offs* e recursos utilizados, a operação da ferramenta nas etapas de registro, autenticação e troca de senha se deu da seguinte maneira:

Inicialização do Servidor

O servidor gera uma chave privada x através do módulo *Crypto.PublicKey* do Pycryptodome utilizando a curva elíptica 'P-256'. Na sequência, o servidor escolhe 2^8 como parâmetro l do verificador difuso. Assim, o servidor computa a chave pública xP.

Registro

Para realizar o registro, primeiramente, o médico usuário U_i seleciona um nome identificador ID_i e um endereço de email e envia ambos para o servidor S. Após receber o requerimento de registro, S gera um número aleatório n_i através do módulo *Crypto.Rand*, computa $K_i = h_i(ID_i||x||n_i)$ com o módulo *Crypto.Hash*, e S produz um código QR QR_i através do módulo *Pyqrcode*, contendo $Ki, xP, l, h_1(.), h_2(.), BH(.)$. Posteriormente, S envia o código QR para o endereço de email definido por U_i através do módulo *email.mime*. Por fim, S armazena ID_i, n_i dos U_i em seu banco de dados, através do módulo *MySQL Database*.

Para ativação do registro, U_i transfere o código QR_i para seu *smartphone* e o apresenta para a câmera com o auxílio de uma janela de retorno de vídeo presente na tela do sistema. As informações em QR_i são decodificadas com o módulo Pyzbar. Então U_i insere o ID_i, a senha PW_i e coleta a biometria Bi.

Durante a coleta da biometria na fase de ativação, uma nova janela de retorno de vídeo será aberta e o usuário U_i é orientado a posicionar sua face com olhar voltado para a câmera. Após o U_i confirmar o posicionamento, são tiradas seis fotos de sua face. Estas fotos são armazenadas em um diretório possuindo como nome o ID_i concatenado com sua inversão espelhada e criptografado com a cifra AES através do módulo *Crypto.Cipher*, utilizando o ID_i espelhado como chave. As imagens do diretório são, então, codificadas pelo módulo *encodefaces*, que atualizará a base de faces codificadas com o nome do diretório de imagens de U_i relacionado a seus vetores de características, e por fim atribuindo a B_i o nome do diretório.

O sistema então computa $V_i = h_2(h_2(ID||PW_i||BH(B_i)) \ mod \ l)$, $K_i' = K_i \oplus h_1(ID_i||PW_i||BH(B_i))$ através do módulo *Crypto.Util*. Por fim, é gerado um

novo Código QR QR_i^2 atualizado com as informações $V_i, K_i', xP, l, h_1(.), h_2(.), BH(.)$. O código é exposto na tela e deve ser fotografado pelo médico usuário U_i com seu *smartphone*, sendo posteriormente apagado do sistema.

Login

Para realizar o *login*, o servidor S registra um número aleatório d_s como chave privada temporária utilizando *ECC* através do módulo *Crypto.PublicKey*. Depois, S, com o mesmo módulo, computa a chave pública temporária d_sP e a envia para U_i.

Em seguida, U_i carrega seu código QR QR_i^2 em seu *smartphone* e o apresenta para câmera, tendo as informações decodificadas. Na sequência, insere sua identidade ID_i, sua senha PW_i e coleta sua biometria B_i. Durante a coleta da biometria de U_i na fase de *login*, uma nova janela com retorno de vídeo é aberta e U_i deverá posicionar sua face voltada para a câmera. Os *frames* capturados serão analisados através do módulo *face recognition*, que caso identifique alguma face conhecida irá atribuir a B_i o nome registrado na base de faces codificadas atribuído àquela face.

Com as informações de QR_i^2, o sistema computa $h_2(h_2(ID_i||PW_i||BH(B_i))\ mod\ l)$, e verifica se tal é igual ao valor V_i. Caso seja, o sistema seleciona por ECC um número aleatório como chave privada temporária d_u e computa d_uP. Posteriormente, computa $K_i = K_i' \oplus h_1(ID_i||PW_i||BH(B_i))$, $AID_i = ID_i \oplus h_1(d_uP||d_ud_sP)$, $SK = h_1(ID_i||K_i||d_ud_sP)$, $M_1 = d_uxP$ $M_2 = h_1(ID_i||d_uP||d_sP||K_i)$, e transmite (AID_i, M_1, M_2) para S. Ao receber essa mensagem, S, primeiramente, computa $d_uP = x^{-1}M_1$ e deriva ID_i por meio da computação de $ID_i = AID_i \oplus h_1(d_uP||d_ud_sP)$. Depois, computa $K_i = h_1(ID_i||x||n_i)$ e compara se $h_1(ID_i||d_uP||d_sP||K_i)$ é igual a M_2. Caso seja igual, S computa $SK = h(ID_i||K_i||d_ud_sP), M_3 = h_1(K_i||d_uP||SK)$. Por fim, S envia (M_3) para U_i.

Com o recebimento da mensagem, U_i calcula $h_1(K_i||d_uP||SK)$ e verifica se é igual a M_3 na mensagem recebida. Caso sim, U_i e S são mutuamente autenticados. Caso não, no próximo *login* bem sucedido de U_i, o usuário receberá uma mensagem informando que possivelmente seu código QR foi violado, e será sugerido que o mesmo realize uma troca de senha.

Troca de Senha

Caso um usuário legal queira mudar sua senha, U_i carrega seu código QR QR_i^2 no seu *smartphone* e o apresenta para a câmera para sua decodificação. U_i, então, submete ID_i, PW_i e B_i. Com as informações de QR_i^2, o sistema verifica se $V_i? = h_2(h_2(ID_i||PW_i||BH(B_i))\ mod\ l)$. Caso sim, U_i seleciona uma nova senha PW_i^{new}, SC_i computa $V_i^{new} = h_2(h_2(ID_i||PW_i||BH(B_i))\ mod\ l)$ e $K_i'^{new} = K_i' \oplus h_1(ID_i||PW_i||BH(B_i) \oplus h_1(ID_i||PW_i^{new}||BH(B_i)$. Por fim, um novo código QR QR_i^{new} é gerado e são feitas as atualizações de V_i com V_i^{new} e K_i com K_i^{new}. O resultado é exposto na tela do sistema e U_i deve realizar seu registro através da câmera de seu *smartphone*. QR_i^{new} é apagado do sistema em seguida.

4.2. *Impressão Digital → Reconhecimento Facial*

A autenticação através do uso de impressões digitais, que é utilizado com *Biohashing* [Lacharme et al. 2013], no qual o código obtido com as impressões é criptografado e, então, utilizado para as comparações de autorização, é capaz de atingir níveis de

segurança que impedem sua clonagem. Contudo, a implementação deste método demanda a aquisição de um leitor de qualidade, que atualmente custa em média R$550,00 nas lojas especializadas.

Uma solução para tal alternativa reside na utilização de reconhecimento facial, através da câmera já presente na arquitetura do sistema. Um ponto negativo do método é o fato de a biometria facial ser mais suscetível a falhas quando comparada com a impressão digital, uma vez que fotos são frequentemente utilizadas para burlar o fator de autenticação. Contudo, como a ferramenta proposta utiliza de outros fatores para um maior controle de acesso, considera-se que tal vulnerabilidade do reconhecimento facial não impedirá que o sistema como um todo seja resistente contra ataques.

O principal problema gerado pela utilização de biometria facial encontra-se especificamente na aplicação do *Biohashing*. Este artigo não tem foco em um nível aprofundado dos algoritmos de *machine learning*, mas a utilização de criptografia nos vetores de características poderia afetar todo o processo de classificação do algoritmo utilizado. Assim, tendo em vista esse obstáculo, optou-se por aplicar uma criptografia pós classificação. Nesta, a classificação de saída do processo de reconhecimento é concatenada com seu inverso e, consequentemente, criptografada. Para o mesmo, utilizou-se do inverso da classificação como chave de criptografia.

5. Avaliação da Ferramenta

5.1. Ambiente de testes

Para o teste da implementação da ferramenta, buscou-se aproximar-se dos recursos presentes no ambiente dos CSVs, utilizando-se os seguintes equipamentos:

- Máquina Virtual (Ubuntu 20.04 LTS, 4GB de RAM e 20 GB de HD)
- Webcam Logitech HD Pro C920
- Smartphone Alcatel Light Pixi 4 (Tela 4 Polegadas, Câmera de 8MP)

5.2. Resultados dos testes

5.2.1. Validação e Impacto dos Trade-offs

O grande ponto de preocupação da implementação dos *trade-offs* girava em torno da capacidade de manter as características de segurança do protocolo de Jiang et al.. Assim, observou-se:

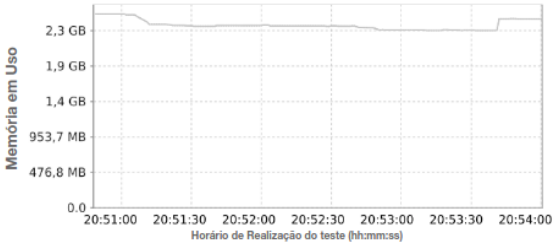
- *Reconhecimento Facial* - A ferramenta não apresentou dificuldade na identificação em faces posicionadas em até 30 cm do dispositivo de captura realizando o reconhecimento em 99,9% dos frames, independente da iluminação ambiente. Já em situações em que o usuário estava posicionado a um metro e meio, ambientes com iluminação direcionada para face obtiveram uma taxa de acerto de 99,5%, apresentando desempenho inferior quando a iluminação ambiente provocava sombreamento na face do usuário realizando o reconhecimento em 69% dos *frames*. Um ponto negativo foi a já prevista falha do algoritmo ao fornecer falsos positivos quando eram apresentadas para a câmera fotos do usuário de interesse. De fato, na maior parte dos testes, o uso da foto foi mais assertivo que o reconhecimento do próprio usuário, devido a iluminação dos *smartphones*. Contudo, cabe observar

que o usuário na outra ponta da comunicação, seja o médico ou o paciente, observaria o uso de uma foto ao invés da própria face para a autenticação, impedindo a realização do ataque.

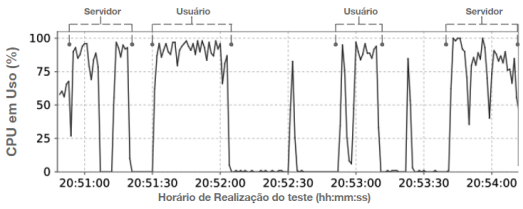
- *Código QR*: Durante os testes a preocupação com os códigos QR estava concentrada nas etapas de ativação de cartão e troca de senhas. Era necessário que até mesmo celulares de entrada de mercado, portadores de *hardwares* mais modestos, fossem capazes de fotografar e reproduzir o código QR com resolução suficiente para que a câmera fosse capaz de realizar sua decodificação no momento de *login*. Os testes com o modelo de smartphone utilizado foram todos bem sucedidos com a câmera sendo capaz de proporcionar a decodificação dos códigos apresentados.

5.2.2. Desempenho Computacional

O Desempenho computacional é outro fator importante, que deve ser levado em consideração na escalabilidade do projeto. As Figuras 2a e 2b detalham, respectivamente, o custo de memória e CPU dos processos do usuário e do servidor nas seções de cadastro, ativação e *login* da ferramenta. Nota-se que o custo de memória manteve-se praticamente constante em torno do baseline de consumo de memória previamente definido de 2GB durante todas as três etapas do processo de autenticação. Já o processamento se mostrou-se mais intenso em etapas de emissão de QR *codes* e de computação das mensagens de autenticação pelo servidor. No usuário, os picos de processamento ocorreram durantes as ativações da câmera.



(a) Uso de memória.



(b) Uso de CPU.

Figura 2. Avaliação de desempenho computacional com o uso da solução proposta no servidor.

5.2.3. Tempo de Resposta ao Usuário

Um indicador importante que permite avaliar a viabilidade da ferramenta pelo ponto de vista do usuário é o tempo de resposta da aplicação. O gráfico presente na Figura 3 exibe o tempo de resposta da aplicação ao usuário em cada uma das etapas da autenticação, sendo as em cinza escuro dependentes única e exclusivamente do sistema, enquanto as em cinza claro são influenciadas pela agilidade do usuário. O teste foi realizado por um usuário com conhecimento do funcionamento da ferramente, simulando um médico habituado com o uso do mesmo, com foco na avaliação da latência gerada pelo sistema.

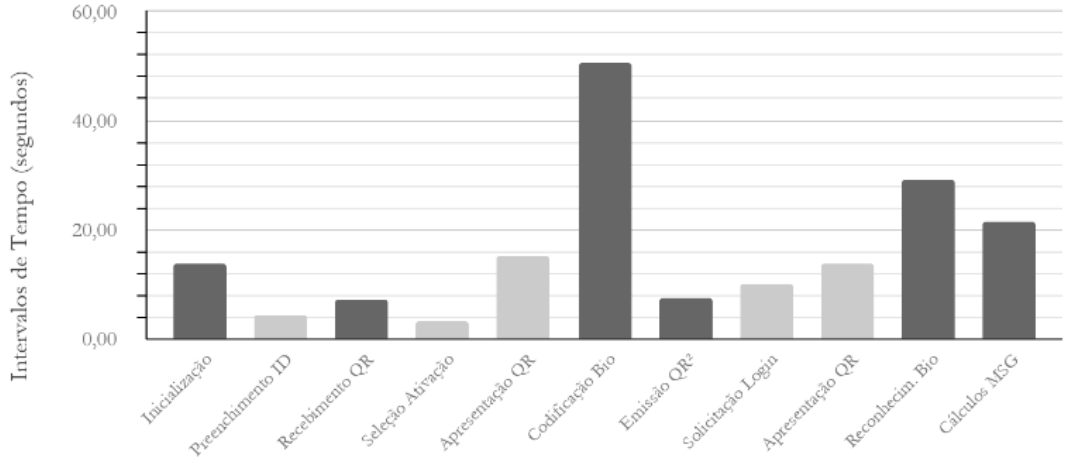


Figura 3. Latência de resposta a cada uma das requisições do usuário durante o processo de autenticação.

6. Conclusão

Neste artigo, foi proposta e implementada uma nova ferramenta de controle de acesso para o Sistema de Telessaúde Hologŕafico da UFF, nomeada STH-HELENA. Primeiramente, a análise dos mais modernos protocolos de autenticação para sistema de telessaúde, levou a escolha e adaptação do protocolo proposto por Jiang et al. para a realidade das arquiteturas de baixo custo dos CSV, gerando dois grandes *trade-offs*.

Na etapa de testes, os dois *trade-offs* apresentaram resultados um pouco distintos. Por sua vez, a utilização de códigos QR foi capaz de atender os requisitos de utilização de celulares de entrada, mostrando grande potencial de aplicação.Além disso, sua implementação não necessitou de nenhuma alteração na logística do protocolo que pudesse comprometer algum dos fatores chave de segurança definidos. Já os testes com o reconhecimento facial, apesar de terem mostrado sua capacidade de realizar o reconhecimento das faces já cadastradas, deixaram evidente a susceptibilidade de quebra deste fator. Apesar disso, devido a presença dos outros fatores de segurança e da própria natureza da aplicação, o impacto desta susceptibilidade torna-se menor.

Dentre as características relativas aos requisitos computacionais da utilização da ferramenta, pode se avaliar com os testes que seus requisitos de *hardware* se aproximam de computadores populares o que permite contemplar um maior contingente de equipamentos já presentes nas infraestruturas das unidades médicas periféricas.

O último teste realizado nos permitiu verificar se a ferramenta seria capaz de prover um tempo de resposta confortável ao usuário durante a permutação de suas etapas. Os resultados obtidos mostraram que a etapa de codificação da biometria tem aproximadamente cinco vezes o tempo de resposta considerado limite para que a aplicação consiga manter o foco do usuário [Nielsen 1993]. Como esta etapa ocorre apenas no momento de ativação do código QR, de forma semelhante a qual outras etapas de alta latência independentes de ação do usuário são poucas vezes utilizadas na etapa de *login*, esta latência excessiva pode ser mitigada com a futura implementação de um *feedback* visual e simultâneo a estas execuções, fornecendo ao usuário garantia do funcionamento da aplicação e a noção de seu tempo de espera.

Por fim conclui-se que a nova ferramenta, STH-HELENA se mostra capaz de manter as características de segurança do protocolo no qual foi baseada, trazendo consigo soluções para sua implementação em infraestruturas de baixo custo. Sendo assim possível, em conjunto com a plataforma de gerência de mídias do STH-UFF, servir como base para expansão do projeto na rede pública de saúde.

Referências

Arshad H and Nikooghadam M (2014). Three-factor anonymous authentication and key agreement scheme for Telecare medicine information systems. *J Med Syst*.

Beaklini, A. and Fonseca et. al, A. (2018). Interiorização da medicina utilizando um sistema de telepresença holográfico. NETAv.

Bello, D. (2016). Implantação e execução de sistema holográfico em centro de saúde da uff vinculado a ações de saúde com a marinha do brasil (projeto telessaúde). Apoio Faperj.

Fonseca, A. (2019). Desenvolvimento de um ambiente seguro para projeção de áudio e vídeo em tempo real para o sistema de telessaúde holográfico da uff. Dissertação apresentada ao Curso de Mestrado em Engenharia de Telecomunicações da Universidade Federal Fluminense. Área de Concentração: Sistemas de Telecomunicações.

Galvão, R. (2006). O problema do logaritmo discreto em curvas elípticas.

Geitgey, A. (2018). Raspberry pi face recognition.

He et al., D. (2015). A secure temporal-creden- tial-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Inf Sci*.

IBGE (2020). Pnad contínua tic 2018: Internet chega a 79,1% dos domicílios do país.

Jiang et al., Q. (2018). Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systemss. *J Ambient Intell Human Comput*, (16):1061—1073.

Lacharme, P., Cherrier, E., and Rosenberger, C. (2013). Preimage attack on biohashing.

Lamport L (1981). Password authentication with insecure communication. *Commun ACM*, pages 770–772.

Lu et al. Y (2014). An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. *J Med Syst*.

NBR27001, A. N. I. (2006). Tecnologia da informação – técnicas de segurança –sistemas de gestão de segurança da informação – requisitos.

Nielsen, J. (1993). Response times: The 3 important limits.

Stallings, W. (2015). *Criptografia e segurança de redes: princípios e práticas*. Pearson Brasil, 6th edition.

Tan Z (2013). An efficient biometrics-based authentication scheme for telecare medicine information systems. *Network*, page 200–204.

Wang, D. (2016). Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound. *IEEE Trans Dependable Secure Comput.*