UNIVERSIDADE FEDERAL DO PAMPA

Vagner Ereno Quincozes

Um Sistema Seguro para Autenticação e Identificação de Técnicos e Clientes de ISP

Alegrete
2021

Vagner Ereno Quincozes

Um Sistema Seguro para Autenticação e Identificação de Técnicos e Clientes de ISP

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. Dr. Rodrigo B. Mansilha

Alegrete
2021

SERVIÇO PÚBLICO FEDERAL

MINISTÉRIO DA EDUCAÇÃO

Universidade Federal do Pampa

VAGNER ERENO QUINCOZES

UM SISTEMA SEGURO PARA AUTENTICAÇÃO E IDENTIFICAÇÃO DE TÉCNICOS E CLIENTES DE ISP

Monografia apresentada ao Programa de Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Monografia defendida e aprovada em 07, de maio de 2021.

Banca examinadora:

_____

Prof. Dr. Rodrigo Brandão Mansilha

Orientador

Unipampa

_____

Prof. Dr. Diego Luis Kreutz

Unipampa

_____

Prof. Dr. Charles Christian Miers

UDESC

Assinado eletronicamente por **RODRIGO BRANDAO MANSILHA**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 12/05/2021, às 09:53, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Charles Christian miers**, **Usuário Externo**, em 12/05/2021, às 10:03, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **DIEGO LUIS KREUTZ**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 12/05/2021, às 10:29, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0523354** e o código CRC **912A7802**.

RESUMO

O processo de suporte técnico dos *Internet Service Provider* (ISP) regionais ainda utiliza amplamente protocolos frágeis no processo de autenticação entre as entidades envolvidas em suporte técnico (clientes, técnicos e gestores). Nesse contexto, são comumente utilizadas carteirinhas físicas (ou virtuais) baseadas em dados estáticos não autenticáveis, que podem ser facilmente roubadas, clonadas ou reproduzidas, para citar um exemplo. Neste trabalho propõe-se um sistema seguro para autenticação e identificação de clientes, técnicos e gestores de ISP. Basicamente, o sistema é composto por (*i*) um conjunto de protocolos de identificação e autenticação, e (*ii*) uma aplicação móvel que implementa os protocolos de maneira amigável aos usuários. O conjunto de protocolos foi avaliado formalmente através da ferramenta Scyther. A viabilidade técnica da aplicação proposta foi demonstrada através de um protótipo funcional.

Palavras-chave: Protocolo de Autenticação. Protocolo de Identificação. Internet Service Provider (ISP)

ABSTRACT

The technical support process of regional Internet Service Providers (ISP) still widely fragile protocols in the authentication process among entities involved in technical support (customers, technicians and managers). In this context, physical (or virtual) cards are based on static, non-authenticable data, which can be easily stolen, cloned or reproduced, to name an example. This work proposes a secure system for authentication and identification of customers, technicians and ISP managers. Basically, the system comprises (i) a set of identification and authentication protocols, and (ii) a user-friendly mobile application that implements the proposed protocols. The set of protocols were formally assessed using the Scyther tool. The technical feasibility of the proposed application was demonstrated through a functional prototype.

Key-words: *Authentication Protocol. Identification Protocol. Internet Service Provider (ISP)*

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

BKE β^* *Bilateral Key Exchange*

CCP Comunicação por campo de proximidade

CIA *Confidentiality - Integrity - Availability*

CPF Cadastro de Pessoa Física

DDoS *Distributed Denial of Service*

DoS *Denial of Service*

E *Encrypt*

HMAC *Hash-based Message Authentication*

IEC *International Electrotechnical Commission*

IMEI *International Mobile Equipment Identity*

IoT *Internet of Things*

ISP *Internet Service Provider*

LAN *Local Area Network*

MD4 *Message Digest Algorithm 4*

MD5 *Message Digest Algorithm 5*

MITM *man-in-the-middle*

MQTT *Message Queue Telemetry Transport*

NFC *Near-Field Communication*

NS α *Needham-Schroeder*

NSA *National Security Agency*

NSL β *Needham-Schroeder-Lowe*

OTAC *One-Time Authentication Code*

OTPs *One-Time Passwords*

PAN *Personal Area Network*

PKE *Public Key Encryption*

PKI *Public Key Infrastructure*

QR Code *Quick Response Code*

RF Requisitos Funcionais

RFID *Radio Frequency IDentification*

RG Registro Geral

RNF Requisitos Não Funcionais

SHA *Secure Hash Algorithm*

SHA-1 *Secure Hash Algorithm 1*

SHA-3 *Secure Hash Algorithm 3*

SMS *Short Message Service*

TEA *Tiny Encryption Algorithm*

UC *User Case*

URI *Uniform Resource Identifier*

VIPER *View-Interactor-Presenter-Entity-Router*

XML *Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

A expansão local e regional dos provedores de serviços de Internet (do inglês, *Internet Service Providers* (ISP)), vem sendo motivada por diferentes fatores, como a personalização no atendimento e a agilidade no suporte técnico (Bettio, 2016). Nesse contexto, os investimentos em agilidade e economicidade acabam sendo priorizados em relação aos investimentos em segurança.

Ao observar o modo de operação de ISPs regionais[1], identificou-se que o processo de autenticação das entidades envolvidas em suporte (*i.e.,* cliente, técnico e gestor de atendimento do ISP) tem sido realizado empregando protocolos e mecanismos frágeis de autenticação, como carteirinhas físicas (ou virtuais) baseadas em dados estáticos não autenticáveis (*e.g.*, foto, nome e código de barras estático). Em primeiro lugar, essas carteirinhas podem ser roubadas, clonadas, ou reproduzidas, comprometendo o processo de autenticação entre as entidades mencionadas anteriormente. Em segundo lugar, um cliente do ISP, atualmente, não dispõe de nenhuma forma para autenticar o técnico, o qual chega até sua casa para realizar o suporte técnico especializado. A carteirinha do técnico costuma ser simples, contendo apenas uma foto, um nome e, em alguns casos, um código de barras estático contendo um dado identificador do técnico, como seu Cadastro de Pessoa Física (CPF).

Na literatura existem propostas de protocolos de autenticação de múltiplas entidades, como *Needham-Schroeder* (NS α), *Needham-Schroeder-Lowe* (NSL β) e *Bilateral Key Exchange* (BKE β^*) (Cremers and Mauw, 2006). Entretanto, esses protocolos tradicionais assumem a existência de uma infraestrutura de chaves públicas (do inglês, *Public Key Infrastructure* (PKI)). Tal abordagem pode ser complexa, indesejada ou desnecessária em cenários como o da gestão da segurança de infraestruturas de redes programáveis (Kreutz et al., 2019). Soluções de autenticação mais recentes, voltadas para aplicativos de dispositivos móveis, como a 2FMA-NetBank (Pratama and Prima, 2016), também dependem de uma PKI e, além disso, adicionam múltiplos fatores de autenticação priorizando aspectos de segurança em detrimento de aspectos de usabilidade.

Recentemente, Kreutz et al. (2020a) propuseram a Auth4App. Tal proposta é composta por protocolos projetados para a autenticação de duas entidades quaisquer (*e.g.,* duas pessoas) utilizando um aplicativo para dispositivos móveis. O Auth4App destaca-se por: (*i*) não depender de uma PKI para realizar o processo de vinculação e autenticação de entidades e (*ii*) não necessitar de múltiplos fatores tradicionais de autenticação (*e.g.,* login/senha e um *Short Message Service* (SMS) com código de verificação) para garantir níveis mais elevados de segurança.

[1] Os nomes dos ISPs foram omitidos por questões de sigilo.

1.1 Objetivo

O objetivo geral deste trabalho é apresentar um sistema seguro para autenticação e identificação de técnicos e clientes de ISP. Nesse sentido, são visados dois objetivos específicos: (*i*) conceber um conjunto de protocolos de autenticação e identificação, e (*ii*) conceber uma aplicação móvel que instancia os protocolos propostos de maneira amigável.

Os protocolos combinam características essenciais de protocolos como o BKE β^* e de soluções recentes como a Auth4App. O protocolo principal da solução proposta, denominado BKE-Auth4ISP (Quincozes et al., 2020a), foi formalmente verificado. Para isso, utilizou-se a ferramenta Scyther (Cremers, 2006b) que indicou que o protocolo é seguro. Para demonstrar a viabilidade técnica do protocolo proposto, desenvolveu-se um protótipo de um aplicativo para dispositivos móveis.

1.2 Organização deste Trabalho

O restante deste trabalho está organizado como segue. O Capítulo 2 aborda os conceitos fundamentais para o entendimento do problema e a concepção e avaliação de uma solução (*e.g.,* propriedades e mecanismos de segurança, a ferramenta Scyther e as tecnologias de comunicação de área pessoal). O Capítulo 3 resume os trabalhos encontrados na literatura considerados mais relacionados com esta monografia. No Capítulo 4, o sistema proposto é explanado, incluindo: casos de uso (Seção 4.1); modelo de ataque (Seção 4.2); protocolos de identificação e autenticação de múltiplas entidades, denominado BKE-Auth4ISP (Seção 4.3); e, uma explicação sobre o aplicativo prototipado (Seção 4.4). O Capítulo 5 apresenta uma verificação formal do protocolo proposto e demonstra a viabilidade técnica da aplicação proposta, através de um protótipo funcional. Por fim, o Capítulo 6 discute as considerações finais, elenca os resultados alcançados e discorre sobre potenciais trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, primeiramente são abordados alguns conceitos fundamentais, como a Segurança da Informação (Seção 2.1) e suas respectivas propriedades. Em seguida, é realizada uma análise das tecnologias de comunicação que possibilitam a execução dos protocolos propostos, proporcionando acesso amigável aos usuários (Seção 2.2).

2.1 Segurança da Informação

Nesta seção são explanadas as Propriedades de Segurança (Subseção 2.1.1), Mecanismos de Segurança (Subseção 2.1.2) e as Ferramentas para Verificação de Protocolos de Segurança (Subseção 2.1.3).

2.1.1 Propriedades de Segurança

Há três critérios principais de segurança da informação, conhecidos pela tríade Confidencialidade, Integridade e Disponibilidade, do inglês *Confidentiality* - *Integrity* - *Availability* (CIA), conforme ilustrado na Figura 1. Tais critérios são responsáveis por garantir, respectivamente, (*i*) que informações não estejam disponíveis a indivíduos não autorizados, (*ii*) exatidão e completeza de ativos, e (*iii*) a propriedade de estar acessível e utilizável quando demandado por uma entidade autorizada (Anand et al., 2020)(Stallings, 2006).

Figura 1 – Tríade CIA.

Fonte: o autor.

- **Confidencialidade** visa restringir os dados, disponibilizando somente para usuários autorizados. A confidencialidade é capaz de proteger as transmissões de dados de ataques passivos. Esses ataques costumam ser utilizados para escutar e monitorar transmissões (Liang et al., 2009), a fim de obter informações diretas (dados em si) e/ou indiretas, para fazer conjecturas (origem, destino, frequência, tamanho ou duração) (Stallings et al., 2008).

- **Integridade** visa garantir que o conjunto de mensagens sejam recebidas conforme foram enviadas, ou seja: sem duplicação, modificação, reordenação ou repetição de mensagens. O critério geral, pode ser especificado, como integridade orientada a conexão, que além de garantir que as mensagens sejam recebidas conforme foram enviadas, trata da negação de serviço. Também, existe a integridade sem conexão, que geralmente oferece proteção somente contra modificações de mensagem (Stallings et al., 2008).

- A **Disponibilidade** diz que um sistema deve oferecer serviços de acordo com o projetado, sempre que uma entidade autorizada solicitar acesso. Ademais, o serviço de disponibilidade deve prover meios de segurança, fornecendo garantia de que estará disponível quando requisitado. Existem ataques que podem resultar na perda e/ou redução de disponibilidade (*e.g. Distributed Denial of Service* (DDoS)). Alguns desses ataques podem ser evitados com métodos de autenticação e criptografia, enquanto outros requerem ação física de um indivíduo (Stallings et al., 2008).

Adicionalmente, há duas propriedades de segurança que podem ser utilizadas: a autenticidade e irretratabilidade. Com a adoção dessas propriedades, a segurança da informação passou a incorporar os cinco Pilares da Segurança da Informação (Cherdantseva and Hilton, 2013).

- **Autenticidade** é uma propriedade de segurança que tem duas principais preoupações: (*i*) garantir que uma comunicação seja autêntica (*i.e.,* autenticidade de conteúdo), e (*ii*) garantir que um agente malicioso não interfira na conexão (*i.e.,* autenticidade de fonte). Em outras palavras, o serviço de autenticação deve fornecer ao destinatário a certeza de que a mensagem é da fonte que afirma ser, e deve assegurar que um terceiro não interfira em uma conexão fingindo ser uma parte legítima (Stallings et al., 2008).

- A **Irretratabilidade** tem como objetivo impedir que o remetente (ou destinatário) negue uma mensagem transmitida (ou recebida). Em suma, o destinatário deve ser capaz de provar que o remetente de fato enviou a mensagem. Similarmente, o remetente deve ser capaz de provar que o destinatário recebeu a mensagem (Stallings et al., 2008).

2.1.2 Mecanismos de Segurança

Para concretizar as propriedades de segurança, é necessário a adoção de mecanismos e métodos de proteção. Nesta seção, alguns serviços e conceitos utilizados para contribuir com esse propósito são apresentados e discutidos.

- **Controle de Acesso** é a capacidade de limitar o acesso de entidades a um recurso, como ao hospedeiro de sistemas e aplicativos por meio de comunicação local ou *links*. As entidades devem estar identificadas e autenticadas para receber o direito de acesso (Stallings et al., 2008).

- ***Nonce*** é um identificador ou número utilizado apenas uma vez, geralmente para impedir ataques de repetição (do inglês, *replay attacks*). Por exemplo, Stallings et al. (2008) define ataques ativos de repetição como a captura passiva de dados e sua respectiva retransmissão para efetivar um efeito não autorizado. Os *nonces* são tipicamente instanciados através de algum gerador de número aleatório, e são conhecidos principalmente em protocolos de segurança. Os *nonces*, quando utilizados em mecanismos de segurança, podem prover unicidade às mensagens. Assim, um ataque de repetição pode ser detectado quando um mesmo *nonce* é recebido em mais de uma mensagem (Patel, 2008)(Quincozes, 2015).

- ***One-Time Authentication Code*** **(OTAC)**, ou códigos de autenticação única, são códigos utilizados para realizar autenticação de forma segura. Os OTACs são conhecidos como códigos descartáveis, pois são válidos para autenticar a identididade de um usuário em apenas uma sessão.

- A **Infraestrutura de Chave Pública (do inglês, *Public Key Infrastructure* (PKI)**) é composta por um conjunto de softwares, hardwares, pessoas, políticas, funções e procedimentos utilizados para gerenciar certificados digitais e criptografia de chave pública (Shirey, 2000). A criptografia de chave pública, também conhecida como criptografia assimétrica, é um sistema criptográfico no qual as partes envolvidas utilizam um par de chaves, composto por uma chave pública e outra privada. As chaves públicas são disponibilizadas para qualquer indivíduo interessado, enquanto as chaves privadas são conhecidas somente pelos seus respectivos proprietários. Com acesso à chave pública do destinatário, qualquer pessoa pode cifrar uma mensagem. No entanto, essa mensagem só pode ser decifrada utilizando a chave privada do destinatário. Para que a segurança seja efetiva nesse esquema, é necessário manter a chave privada segura (Stallings, 2006).

- A **Criptografia de Chave Simétrica**, também conhecida como criptografia de chave secreta ou chave privada, consiste na utilização de uma mesma chave para codificação de texto às claras e decodificação de texto cifrado. Na prática, a criptografia de chave simétrica representa um segredo comum compartilhado entre as

partes envolvidas, necessitando que ambas as partes tenham acesso a mesma chave secreta para poder acessar tais informações. Estudos recentes demonstram que a chave simétrica apresenta menor sobrecarga computacional que algoritmos de chave assimétrica (Williams et al., 2020).

- **Função de Resumo (*Hash*)** é um método criptográfico que gera um resultado único e de tamanho fixo. Na criptografia, uma função *hash* permite verificar rapidamente as entradas de dados recebidas para um valor de *hash* fornecido, mas, caso os dados de entrada sejam desconhecidos, é extremamente difícil reconstruí-los conhecendo somente o valor *hash* armazenado. Isso pode ser utilizado para construção de ***Hash-based Message Authentication* (HMAC)**, que é um tipo de código de autenticação de mensagem que envolve uma função *hash* e uma chave assimétrica. Os HMACs são utilizados para verificar, ao mesmo tempo, a integridade dos dados e autenticidade das mensagens (Stallings, 2006).

  Existem diversos tipos de funções *hash*. Rivest (1990) criou a função *hash Message Digest Algorithm 4* (MD4) em 1990, a qual foi aprimorada pelo autor em 1992, dando origem ao *hash Message Digest Algorithm 5* (MD5) (Rivest and Dusse, 1992). Ambos são resumos *hash* com tamanho de 128-bit. Vale ressltar que, atualmente, o MD5 é considerado completamente vulnerável, podendo sofrer fortes ataques de colisão (Savage, 2008). Em 1993, a *National Security Agency* (NSA) publicou uma nova função, muito similar ao MD5, chamada *Secure Hash Algorithm* (SHA). Mais tarde, em 1995, a NSA descobriu uma falha no SHA e lançou uma nova versão do algoritmo, denominada *Secure Hash Algorithm 1* (SHA-1), com tamanho de 160-bit. Em 2012, devido aos problemas de colisão na família MD e com o objetivo de substituir seus antecessores SHA-1 e SHA-2, surgiu o *Secure Hash Algorithm 3* (SHA-3) (Dworkin, 2015). Atualmente, a função *hash* mais popular é o SHA-1. Também, existem padrões governamentais de funções *hashs* mais longas e mais difíceis de quebrar, como SHA-224, SHA-256, SHA-384 e SHA-512 (Dworkin, 2015).

2.1.3 Ferramentas para Verificação de Protocolos de Segurança

Existem várias ferramentas produzidas especialmente para auxiliar na verificação formal e automática de protocolos de segurança (Kreutz et al., 2020b). Dentre elas, os autores destacam Scyther (Cremers, 2006b), CryptoVerif (Blanchet, 2008), AVISPA (Armando et al., 2005), ProVerif (Blanchet et al., 2018) e Tamarin Prover (Meier et al., 2013).

Neste trabalho, optou-se por utilizar a ferramenta Scyther, que permite a verificação automática de diferentes protocolos de segurança, tornando-se útil para encontrar vulnerabilidades de segurança. Optou-se pelo uso dessa ferramenta devido às suas características, como: eficiência, caracterização de protocolos, representando todos os possíveis

comportamentos do mesmo, e possibilidade de verificar protocolos com sessões e *nonces* ilimitadas (Cremers, 2008). Além disso, a documentação oficial do Scyther[1] apresenta alguns exemplos de cenários em que a ferramenta foi adotada com êxito, por exemplo, a análise dos conjuntos de protocolos IKEv1 e IKEv2 (Cremers, 2011) e da família de protocolos de autenticação *ISO / IEC* 9798 (Basin et al., 2013), além de ser utilizada para encontrar ataques multiprotocolo em protocolos existentes (*i.e., Needham-Schroeder* e *Kao-Chow*) (Cremers, 2006a).

2.2 Protocolos de Rede de Área Pessoal

No sistema proposto, há troca de dados entre duas ou mais entidades fisicamente próximas, formando uma Rede de Área Pessoal (do inglês, *Personal Area Network* (PAN)) (Zimmerman, 1996). Assim, torna-se necessário a adoção de uma tecnologia que forneça esse serviço. Dentre as opções, destacamos as tecnologias *Quick Response Code* (*QR Code*) e *Near-Field Communication* (NFC), que são capazes de conectar dispositivos ao alcance de um indivíduo em comunicação de curta distância.

- ***QR Code*** é um código de barras bidimensional que pode ser escaneado pela maioria dos celulares equipados com câmera. Esse código é convertido para texto, o qual é capaz de fornecer um endereço *Uniform Resource Identifier* (URI), uma localização georreferenciada, e-mail, SMS ou um contato, para citar alguns exemplos. Há diferentes tamanhos e níveis de *QR Code*, que determinam a quantidade de dados que podem ser armazenados. A Norma *International Electrotechnical Commission* (IEC) 18004 padroniza os tamanhos de *QR Code*, fornecendo 40 versões, sendo que cada versão representa uma capacidade de armazenamento (Wave, 2015). Apesar da linguagem do *QR Code* ser complexa para humanos, computadores e máquinas conseguem decodificá-la facilmente. O uso de *QR Code* tende a aumentar cada vez mais, visto que *smartphones* modernos são equipados com câmeras (Marktscheffel et al., 2016).

- **Comunicação por campo de proximidade (CCP), do inglês *Near-Field Communication* (NFC)**, permite a troca de dados sem fio por aproximação entre dispositivos (*e.g., smartphones*, *smartwatches*, cartões de crédito, crachás, etc.) que possuam um *chip* específico integrado. A tecnologia de comunicação sem fio utiliza ondas de rádio de alta frequência, tradicionalmente a $13.56MHz$ com um alcance próximo de 10 centímetros. A taxa de transmissão de dados pode ser de 106, 216 ou 424 $Kbps$ (Mulliner, 2009). A comunicação por meio da tecnologia é simples: basta que os dispositivos compatíveis estejam próximos (dentro de alcance) para acontecer a troca de informações. A Norma IEC 18092 (for Standardization/International

[1] Disponível em https://people.cispa.io/cas.cremers/scyther/

Electrotechnical Commission et al., 2004) define formas de operação e requisitos para modulação. Haselsteiner and Breitfuß (2006) realizaram um estudo sobre a segurança do NFC. Atualmente, o NFC possui aplicações em diversas áreas, como na saúde (Quincozes and Kazienko, 2016), transporte (Thammarat, 2020) e bancária (Bojjagani and Sastry, 2019).

3 TRABALHOS RELACIONADOS

Até onde se sabe, este é o primeiro trabalho que propõe um sistema seguro para autenticação e identificação de técnicos e clientes de ISP. Diante disso, neste capítulo discute-se os trabalhos mais relacionados com protocolos de segurança para autenticação, ainda que em outros contextos. A Tabela 1 resume os trabalhos considerados mais relacionados, classificados por ordem cronológica, e destacando o trabalho resultante desta monografia. Em seguida, cada trabalho é detalhado.

Tabela 1 – Contextos e contribuições.

Referência	**Contexto**	**Contribuções**
(Cremers and Mauw, 2006)	Não definido	Apresentam uma família de protocolos de autenticação entre múltiplas entidades e propõem seis novos protocolos.
(Koschuch et al., 2013)	*Smartphones*	Propõem um método de autenticação utilizando a presença de um *token* de *hardware* conectado por *bluetooth* para autenticar um usuário.
(Pratama and Prima, 2016)	*Internet Banking*	Propõem um esquema de autenticação mútua de dois fatores para *Internet Banking*.
(Marktscheffel et al., 2016)	Casa inteligente	Propõem um protocolo de autenticação mútua baseado em *QR Code* para *Internet of Things* (IoT).
(Jeong and Cho, 2017)	Hospitais e outras corporações	Projetam e implementam um método de autenticação para gerenciar informações.
(Aghili et al., 2019)	Sistemas médicos da IoT	Analisam a segurança de um protocolo para demonstrar vulnerabilidades e apresentam um sistema de autenticação mútua baseado em *Radio Frequency IDentification* (RFID) para IoT.
(Junior et al., 2019)	*Broker* MQTT	Propõem um protocolo baseado na autenticação mútua entre Publicadores e *Brokers*.
(Chandrakar et al., 2019)	Rede ad hoc veicular	Propõem uma nova versão de um protocolo que era originalmente suscetível a falhas.
(Kreutz et al., 2020a)	*Smartphones*	Propõem um esquema de autenticação flexível para aplicativos de *smartphones*.
(Quincozes et al., 2020a)	**ISP**	**Propõem um conjunto de protocolos de autenticação e identificação para entidades envolvidas em suporte de ISP.**
(Wu et al., 2021)	Redes heterogêneas	Propõem um esquema de gerenciamento de acesso unificado com *One-Time Passwords* (OTPs).
(Lei et al., 2021)	*Smartphones*	Apresentam vulnerabilidades em APIs de sistemas operacionais e propõem um mecanismo de autenticação baseado em SMS.

Fonte: o autor.

- Cremers and Mauw (2006) apresentam uma família de protocolos de autenticação entre múltiplas entidades e discutem seis novos protocolos. Os três primeiros protocolos apresentados são, respectivamente, o protocolo NS generalizado de chave

privada (α), NSL generalizado de chave pública (β) e o protocolo generalizado de troca de chaves bilateral (β^*). Para tornarem-se pequenos e eficientes, os protocolos utilizam autenticação delegada, que significa que as partes envolvidas confiam uma nas outras para autenticar-se. Além disso, os autores propõem uma versão reforçada sem utilização de autenticação delegada para cada um dos protocolos. Para atingir as propriedades de segurança desejadas com a presença de um intruso Dolev-Yao, os autores utilizam quantidade de mensagens $2p - 1$ onde p é o número de entidades, e argumentam que este é o número mínimo de mensagens para prover segurança.

- Koschuch et al. (2013) propõem uma alternativa para métodos comuns de autenticação, como PINs e padrões. O mecanismo utiliza um *token* de *hardware* conectado ao *bluetooth* para autenticar usuários. Segundo os autores, a vantagem do método proposto, em relação aos demais existentes, é a usabilidade, incluindo aspectos como facilidade de uso e desempenho. Uma avaliação de desempenho do mecanismo permitiu demonstrar que o processo de autenticação pode ser concluído em menos de um segundo.

- Pratama and Prima (2016) propõem um esquema de autenticação para *Internet Banking*, denominado 2FMA-NetBank. O esquema proposto utiliza quatro objetos: pares de chaves de criptografia de chaves *Public Key Encryption* (PKE), número do *International Mobile Equipment Identity* (IMEI) de *smartphone*, *token* de software e aplicativo *QR Code*. Resumidamente, o primeiro fator utiliza usuário e senha para login. As chaves PKE funcionam como segundo fator de autenticação e também como autenticação mútua. O IMEI do *smartphone* também é considerado segundo fator de autenticação. Assim, o *token* é instalado no dispositivo do indivíduo a fim de gerar um mecanismo de desafios e respostas. Por fim, o *QR Code* atua como facilitador do usuário no processo de autenticação. Os autores assumem que o servidor do banco armazena a segurança das chaves públicas e privadas e do IMEI do *smartphone*. Segundo os autores, as vantagens dessa solução consistem na eficiência da solução e baixo custo para implementação.

- Marktscheffel et al. (2016) apresentam requisitos para um cenário de casa inteligente, que pode ser adaptado para outros cenários. Para atender os requisitos, propõe-se um protocolo de autenticação mútua baseado em *QR Code*. A abordagem é utilizada para registrar um novo dispositivo em um servidor, melhorando a usabilidade e segurança durante o registro.

- Jeong and Cho (2017) apresenta o projeto e implementação de um método de autenticação que pode ser utilizado para gerenciar informações de hospitais e outras corporações. Além disso, o algoritmo é capaz de autenticar usuários em tempo real utilizando IMEI de telefone inteligente, *QR Code*, BLE e mensagem *push*.

- Aghili et al. (2019) analisam a segurança de um protocolo denominado LRMI e demonstram suas vulnerabilidades. Ademais, propõem um protocolo de autenticação mútua baseado em RFID. Segundo os autores, o protocolo fornece comunicação segura e leve para sistemas médicos da IoT.

- Junior et al. (2019) propõem um mecanismo denominado *LegitimateBroker* para mitigar ataques de personificação em um dispositivo *Broker Message Queue Telemetry Transport* (MQTT). Os autores propõem um protocolo baseado na autenticação mútua entre Publicadores e *Brokers*, o qual fundamenta-se no uso de mecanismos de baixa sobrecarga computacional, tais como funções *hash*, *nonces* e chaves simétricas. Utilizam armazenamento indireto de chaves no *Broker* e renovação periódica de chaves no *Broker* e no Publicador. O mecanismo proposto pelos autores apresentou menor sobrecarga, quando comparado a outras abordagens tradicionais.

- Considerando a importância de proteger a privacidade de usuários em uma rede *ad hoc* veicular, Chandrakar et al. (2019) propõem uma nova versão de um protocolo de autenticação que está suscetível a falhas de segurança. Adicionalmente, os autores fornecem uma prova formal para o protocolo proposto com a finalidade de demonstrar que é seguro.

- Kreutz et al. (2020a) propõem o Auth4App, uma solução de autenticação flexível para aplicativos de *smartphones*. O Auth4App oferece autenticação de fator único baseado em esquema de OTAC, vinculando a identidade de um usuário a apenas um *smartphone*. O Auth4App é relativamente mais simples que os sistemas de múltiplos fatores, como 2FMA-NetBank, pois emprega fator único e é mais seguro que sistemas de fator único convencionais, como carteirinha digital, pois emprega OTAC (*i.e.*, código roubado tem impacto restrito). O Auth4App é composto por dois protocolos: (*i*) vinculação de credenciais do usuário a um único dispositivo móvel, e (*ii*) geração de códigos de autenticação descartáveis. O primeiro protocolo, de vinculação, gera uma chave mestra. Enquanto a chave mestra for válida não será possível vincular o mesmo usuário a outro dispositivo. O segundo protocolo, de autenticação, utiliza a chave mestra para derivar códigos de autenticação únicos OTAC empregados em cada procedimento.

- Wu et al. (2021) fornecem um novo esquema de gerenciamento de acesso unificado, com senhas de uso único (OTPs) baseado nas extensões de guarda de software (do inglês, *Software Guard Extensions*) Intel (SGX). Além disso, segundo os autores, o esquema fornece resistência a ataques *man-in-the-middle* (MITM), ataques de phishing, ataques de repetição e ataques de *Denial of Service* (DoS).

- Lei et al. (2021) revelam vulnerabilidades na autenticação baseada em SMS OTPs. Primeiramente, apresentam falhas de segurança em mecanismos oferecidos por sis-

temas operacionais (e.g. APIs Android), que são desenvolvidos especificamente para proteger contra ameaças de invasores locais. Ademais, constataram que os aplicativos Telegram e KakaoTalk são vulneráveis. Por fim, os autores propõem melhorias e modificações nas APIs desses sistemas operacionais para implementar um novo mecanismo de autenticação baseado em SMS.

Note-se que, dentre os trabalhos relacionados elencados, não há propostas para identificação e autenticação das entidades envolvidas especificamente em suporte técnico (*i.e.,* Cliente, Técnico e Gestor) de ISP regionais. Para preencher essa lacuna, este trabalho propõe um sistema seguro composto por um conjunto de protocolos e uma aplicação móvel.

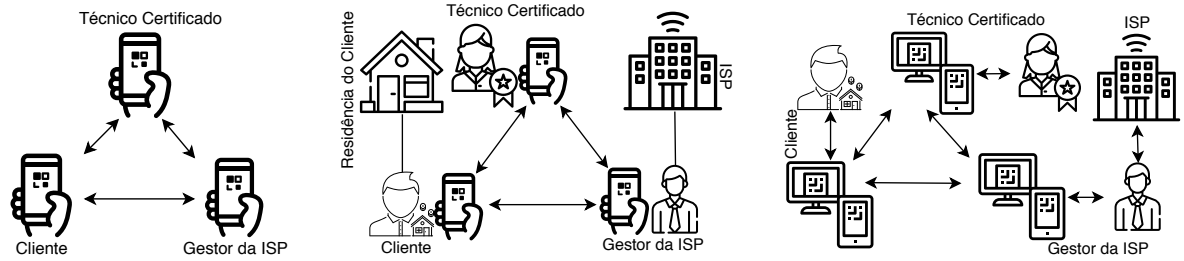
4 SISTEMA PROPOSTO

O sistema proposto neste trabalho é composto por (*i*) um conjunto de protocolos de identificação e autenticação, e (*ii*) uma aplicação móvel. Este capítulo está organizado como segue. A Seção 4.1 apresenta os casos de uso (do inglês, *User Case* (UC)). A Seção 4.2 explana o modelo de ataque considerado. Na Seção 4.3, os protocolos de identificação e autenticação são expostos. Por fim, a Seção 4.4 apresenta uma análise de requisitos da aplicação móvel proposta.

4.1 Casos de Uso

O cenário de autenticação mútua ilustrado na Figura 2(a) pode ser aplicado em dois casos de uso: ambientes físicos e ambientes virtuais. O primeiro caso trata-se de autenticação presencial, isto é, quando os dois indivíduos principais (cliente e técnico) estão no mesmo ambiente físico, como a residência do cliente, gerenciados por um gestor remoto. A Figura 2(b) ilustra a visita de técnico certificado para prestar atendimento a um cliente do ISP. Ao chegar no local do atendimento, o técnico deve apresentar o seu código de identificação, que pode ser representado através de um *QR Code* contendo uma imagem, dados pessoais do técnico, dados do atendimento e um código de autenticação. O cliente do ISP utilizará o seu *smartphone* para ler o *QR Code* e verificar se a pessoa é de fato o técnico certificado indicado por um gestor do ISP. Similarmente, o técnico poderá ler o *QR Code* do cliente para verificar se ele é realmente o cliente correto.

Figura 2 – Casos de Uso da solução de autenticação proposta.



(a) Autenticação da Tríade. (b) Ambiente Físico. (c) Ambiente Virtual.

Fonte: o autor.

A segunda forma de autenticação mútua, conforme ilustrado na Figura 2(c), refere-se ao acesso remoto de suporte. Nesse caso, o técnico certificado irá prestar um atendimento remoto a um cliente do ISP. De maneira análoga ao primeiro cenário, o cliente e o técnico poderão autenticar-se mutuamente. Contudo, diferentemente do primeiro caso, os dados de verificação serão lidos de maneira automática pelo aplicativo, ou através da tela do computador (ou outro dispositivo) com acesso à Internet, dispensando que os indivíduos estejam no mesmo ambiente físico. Em ambos os casos, assume-se que todos os participantes possuem conexão com a Internet.

4.2 Modelo de Ataque

Neste trabalho, assume-se que as ISPs são responsáveis por identificar e cadastrar novos usuários (clientes, técnicos e gestores). Essa premissa é razoável uma vez que, por exemplo, o cliente (tipicamente) estabelece uma relação de confiança ao assinar o contrato de prestação de serviços com o seu ISP. Como implicação, na solução proposta (Seção 4.3), considera-se que os clientes, técnicos e gestores já estão devidamente identificados pelo ISP. Na sequência, são discutidas as premissas particulares para cada um dos casos de aplicação da solução investigada: ambientes físico e virtual.

No ambiente físico, presume-se que o *smartphone* utilizado pelo usuário, seja cliente, técnico ou gestor, é confiável. Essa premissa se justifica pois em caso de substração do dispositivo autenticado, o usuário deverá notificar a ISP. Em seguida, a ISP deverá revogar a identificação do dispositivo e, assim, impedir (o mais breve possível) que um atacante possa se beneficiar do dispositivo subtraído para efetuar ataque de personificação.

No ambiente virtual, o atacante pode realizar ataques de personificação mais sofisticados a partir de acesso à rede. Especificamente, as capacidades do atacante foram modeladas seguindo o modelo Dolev-Yao (Dolev and Yao, 1983). Assim, um intruso poderá controlar a rede e decifrar as mensagens se e somente se conhecer a chave de criptografia. Além disso, o atacante pode modificar, atrasar e inserir comunicação nas mensagens entre usuários (Cremers and Mauw, 2006).

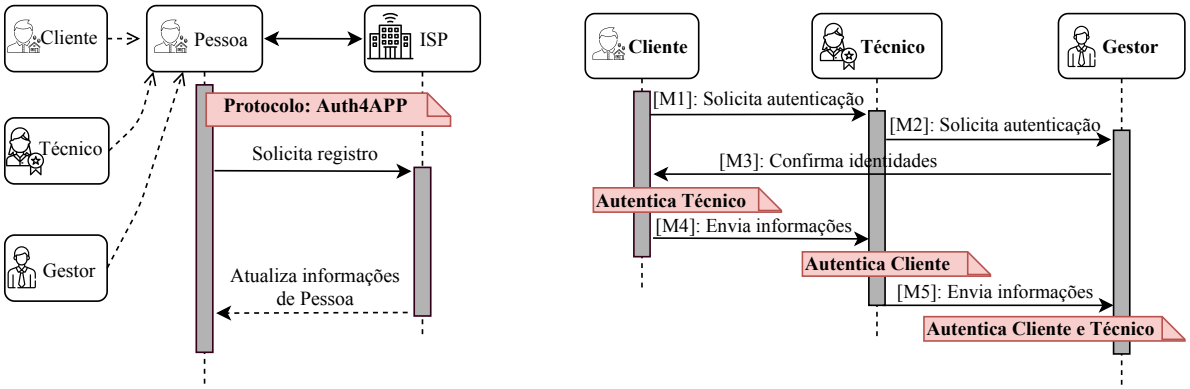
4.3 Protocolos de Identificação e Autenticação

A Figura 3 ilustra o conjunto de protocolos proposto para autenticar o `Cliente`, `Técnico` e `Gestor` do ISP. Tais entidades são consideradas instâncias da entidade `Pessoa`. Cada uma dessas entidades deve ser devidamente identificada no ISP. Posteriormente, a cada atendimento é realizado um processo de autenticação entre cada uma dessas três instâncias.

Para identificar as pessoas que futuramente serão envolvidas em atendimento especializou-se os protocolos apresentados em (Kreutz et al., 2020a), conforme ilustrado na Figura 3(a). A pessoa pode ser qualquer entidade da tríade cliente-técnico-gestor, dependendo do cenário. A Figura 3(b) resume o protocolo de autenticação, denominado *Bilateral Key Exchange for ISP* (BKE4ISP). Esse protocolo é uma instância do BKE* – o menos computacionalmente custoso proposto por Cremers and Mauw (2006) e suficientemente seguro para o modelo de ataque considerado neste trabalho. Além de especificar as entidades em termos de quantidade ($n = 3$) e papéis, BKE4ISP emprega o Auth4App como mecanismo alternativo a uma infraestrutura PKI clássica.

No protocolo, apresentado na Tabela 2, realizou-se duas modificações fundamentais em relação ao protocolo BKE original. Primeiro, utilizou-se OTAC, como proposto na solução Auth4App, e assim a necessidade de uma infraestrutura PKI clássica foi elimi-

Figura 3 – Conjunto de protocolos para autenticação de atendimento do ISP.



(a) Protocolo Auth4ISP. (b) Protocolo BKE4ISP.

Fonte: o autor.

nada. Com OTACs, é possível utilizar apenas algoritmos de cifra simétrica, economizando energia, aumentando o desempenho e assegurando a resistência a ataques de computadores quânticos, como pode ser visto em pesquisas recentes (Kreutz et al., 2019). Segundo, adicionou-se um código de autenticação de mensagem para acelerar o processo de verificação da autenticidade de cada mensagem. O custo computacional de um HMAC é significativamente inferior ao de um algoritmo de cifra (*e.g.,* ver gráfico de desempenho das comunicações com apenas HMAC ou com a adição de algoritmos de cifra (Kreutz et al., 2019)). O desempenho pode impactar na resiliência do protocolo em situações de ataques de negação de serviço (do inglês, *Denial of Service* (DoS)), por exemplo.

Tabela 2 – Protocolo BKE4ISP.

M1	Cliente → Técnico	[E(Cli, Isp, nonce-cli)]HMAC
M2	Técnico → Gestor do ISP	[E(Tec, Cli, nonce-cli, nonce-tec)]HMAC
M3	Gestor do ISP → Cliente	[E(nonce-cli, nonce-tec, nonce-isp, Tec, Isp)]HMAC
M4	Cliente → Técnico	[E(nonce-isp,nonce-tec)]HMAC
M5	Técnico → Gestor do ISP	[E(nonce-isp)]HMAC

Fonte: o autor.

Na prática, para utilizar os OTACs, um cliente pode ler os dados de identificação contidos em um *QR Code* do aplicativo do técnico. O processo é análogo a uma verificação manual dos documentos de identificação do técnico, como carteirinha física ou Registro Geral (RG). Os OTACs são definidos aos pares, ou seja, entre o cliente e o técnico, entre o técnico e o gestor do ISP e entre o gestor do ISP e o cliente. A inicialização do gerador desses códigos dinâmicos e únicos ocorre durante o cadastro e a vinculação ao aplicativo de cada entidade junto ao ISP.

Observa-se no protocolo BKE4ISP, que a ordem das pessoas (*e.g.* Cliente, Técnico

ou Técnico, Cliente) não altera o resultado final do protocolo. A Tabela 2 apresenta o protocolo iniciado pelo Cliente. Nele, todas as mensagens entre duas entidades quaisquer (*e.g.,* cliente-técnico e técnico-gestor do ISP) são cifradas utilizando uma função *Encrypt* (E), cuja chave secreta é um OTAC. O protocolo inicia com o cliente enviando (linha 1) uma mensagem para o técnico contendo um *nonce* de autenticação (`nonce-cli`), sua identificação (`Cli`) e a identificação do gestor do ISP (`Isp`). O técnico acrescenta o seu identificador (`Tec`) e o seu *nonce* de autenticação (`nonce-tec`) e envia para o gestor do ISP (linha 2). O gestor do ISP envia uma mensagem ao cliente (linha 3) contendo os *nonces* de autenticação recebidos, confirmando assim as respectivas identidades, mais o `nonce-isp` do ISP, que é utilizado para finalizar o processo de autenticação mútua entre o cliente e o técnico. O cliente recebe a mensagem do gestor do ISP (linha 3), autentica o técnico e envia os `nonce-isp` e `nonce-tec` para o técnico. O técnico verifica o *nonce* de autenticação recebido, autenticando o cliente, e envia o `nonce-isp` para o gestor do ISP, que autentica o cliente e o técnico, finalizando o processo de autenticação.

4.4 Análise da Aplicação Móvel

Nesta seção, realizou-se uma análise da aplicação móvel proposta. Inicialmente, são explanados os componentes necessários, e em seguida, é realizada uma análise de requisitos funcionais e não funcionais. Por fim, o escopo, os usuários e seus papéis são definidos.

4.4.1 Componentes

Componentes necessários para implementação e execução da solução proposta.

- ***Smartphones*** são utilizados para manter a base de dados e realizar identificação e autenticação. Esses dispositivos proporcionam que o cliente, o técnico e o gestor do ISP modifiquem e recuperem dados do servidor.

- ***QR Code*** transmitem informações de identificação e autenticação dos usuários, as quais são armazenadas no servidor e mantidas pelo gestor do ISP. Em um atendimento, por exemplo, o *QR Code* proporciona aos usuários acesso a essas informações de forma amigável, rápida e segura.

- **Acesso à Internet**. Conforme descrito nos casos de uso (Seção 4.1), assume-se que os indivíduos envolvidos no suporte técnico possuem conexão à Internet.

- **Servidor** é responsável por armazenar registros de clientes, técnicos e gestores do ISP. Por exemplo, o servidor pode armazenar a certificação de um técnico, e as informações pessoais de cada usuário.

- **Aplicação para Plataforma Android**. Permite que os clientes, técnicos e gestores do ISP identifiquem-se e autentiquem-se uns aos outros de maneira amigável.
- **Protocolo de Identificação**. Exposto na (Seção 4.3), serve para que os usuários envolvidos em um atendimento consigam se identificar e provar que são quem dizem ser.
- **Protocolo de Autenticação**. Permite que os usuários envolvidos em atendimentos se autentiquem de forma rápida e segura. O protocolo provê autenticação mútua entre três entidades.

4.4.2 Requisitos Funcionais

A proposta envolve os seguintes Requisitos Funcionais (RF).

- **RF 01**. O software deve permitir que o cliente autentique um técnico.
- **RF 02**. O software deve permitir que o cliente avalie um atendimento prestado por um técnico.
- **RF 03**. O software deve permitir que o técnico autentique um cliente.
- **RF 04**. O software deve permitir que o gestor identifique clientes e técnicos.
- **RF 05**. O software deve permitir que o gestor autentique clientes e técnicos.
- **RF 06**. O gestor deve ser capaz de adicionar novos clientes e técnicos.
- **RF 07**. O gestor deve ser capaz de alterar informações de clientes e técnicos.
- **RF 08**. O gestor deve ser capaz de visualizar informações de clientes e técnicos.
- **RF 09**. O gestor deve ser capaz de excluir clientes e técnicos.
- **RF 10**. O gestor deve ser capaz de revogar a identificação de clientes e técnicos.

4.4.3 Requisitos Não Funcionais

A proposta deve cumprir os seguintes Requisitos Não Funcionais (RNF).

- **RNF 01**. O cliente deve ser capaz de operar o sistema a partir de um *smartphone*.
- **RNF 02**. O cliente deve ser capaz de autenticar um técnico via tecnologia *QR Code* quando utilizar um dispositivo móvel.
- **RNF 03**. O cliente deve ser capaz de avaliar um atendimento recebido por um técnico utilizando um dispositivo móvel.

- **RNF 04**. O técnico deve ser capaz de operar o sistema a partir de um *smartphone*.
- **RNF 05**. O técnico deve ser capaz de autenticar um cliente via tecnologia *QR Code* quando utilizar um dispositivo móvel.
- **RNF 06**. O gestor deve ser capaz de realizar todas as operações no sistema a partir de um *smartphone*.
- **RNF 07**. O gestor deve ser capaz de autenticar clientes e técnicos via tecnologia *QR Code* quando utilizar um dispositivo móvel.
- **RNF 08**. O software deve prover confidencialidade de dados[1].
- **RNF 09**. O software deve prover a integridade de dados e da fonte.
- **RNF 10**. O software deve prover irretratabilidade.
- **RNF 11**. O software deve prover autenticidade.
- **RNF 12**. O software deve prover disponibilidade.

4.4.4 Escopo, Usuários e Papéis

Esta seção apresenta detalhadamente o fluxo de atendimentos do ISP, bem como os usuários envolvidos e os papéis de cada um.

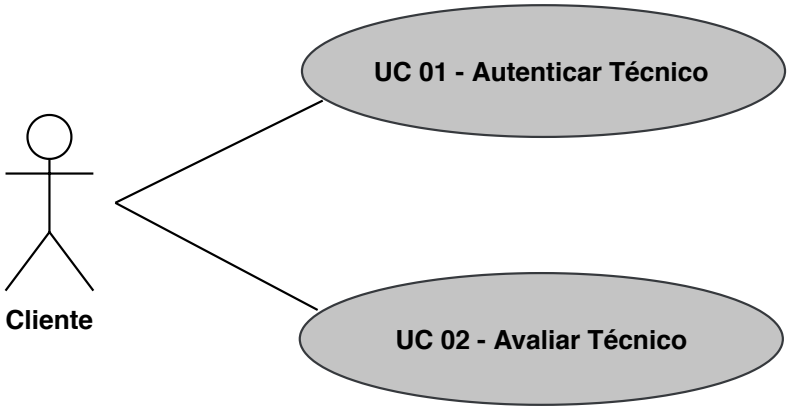
4.4.4.1 Cliente

O cliente é a pessoa que, por exemplo, está com problemas na sua rede de área local, ou *Local Area Network* (LAN) conectada à Internet. Para resolver o problema, o cliente solicita suporte para o ISP, que irá designar um técnico cadastrado na base de dados para atendê-lo presencialmente ou remotamente. No atendimento presencial, por exemplo, assim que o técnico chega ao local de atendimento (*e.g.,* residência do usuário), ele apresenta a sua identificação. O cliente poderá autenticar o técnico para provar que ele é realmente a pessoa encaminhada pelo ISP. Opcionalmente, após o atendimento, o cliente pode avaliar o técnico que foi até a sua residência, fornecendo ao ISP um *feedback* sobre o serviço prestado pelo técnico.

As funcionalidades do cliente são representadas pelos Casos de Uso, do inglês UC 01 e 02 (Figura 4), os quais são descritos nas Tabelas 3 e 4, respectivamente.

[1] Embora este e os próximos RNFs possam ser considerados RFs na literatura de segurança, optou-se por classificá-los como RNF, considerado mais usual na literatura de Engenharia de Software.

Figura 4 – Casos de Uso do Cliente.



Fonte: o autor.

Tabela 3 – Caso de Uso 01 - Autenticar Técnico.

Identificação do Caso de Uso: UC 01 - Autenticar Técnico
Ator Principal: Cliente
Pré-condições: O usuário deve estar logado como "Cliente".
Requisitos Funcionais
RF 01. O software deve permitir que o cliente autentique um técnico.
Requisitos Não Funcionais
RNF 01. O cliente deve ser capaz de operar o sistema a partir de um *smartphone*.
RNF 02. O cliente deve ser capaz de autenticar um técnico via tecnologia *QR Code* quando utilizar um dispositivo móvel.

Fonte: o autor.

Tabela 4 – Caso de Uso 02 - Avaliar Técnico.

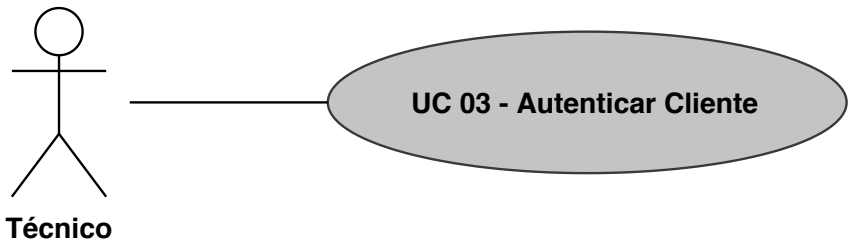
Identificação do Caso de Uso: UC 02 - Avaliar Técnico
Ator Principal: Cliente
Pré-condições: 1. O usuário deve estar logado como "Cliente". 2. O cliente deve ter recebido um atendimento.
Requisitos Funcionais
RF 01. O software deve permitir que o cliente avalie um atendimento prestado por um técnico.
Requisitos Não Funcionais
RNF 01 - O cliente deve ser capaz de operar o sistema a partir de um *smartphone*.
RNF 03 - O cliente deve ser capaz de avaliar um atendimento recebido por um técnico utilizando um dispositivo móvel.

Fonte: o autor.

4.4.4.2 Técnico

O técnico é a pessoa responsável por prestar um atendimento presencialmente ou remotamente a um cliente. Assume-se que os técnicos estão previamente cadastrados e homologados pelo ISP, o qual tem o papel de mantê-los atualizados. Portanto, na execução de um atendimento, o técnico terá uma funcionalidade principal: autenticação do cliente. Isso proporciona ao técnico a certeza de que está atendendo o cliente correto. A funcionalidade principal é descrita pelo UC 03 (Figura 5), exposto na Tabela 5.

Figura 5 – Casos de Uso do Técnico.



Fonte: o autor.

Tabela 5 – Caso de Uso 03 - Autenticar Cliente.

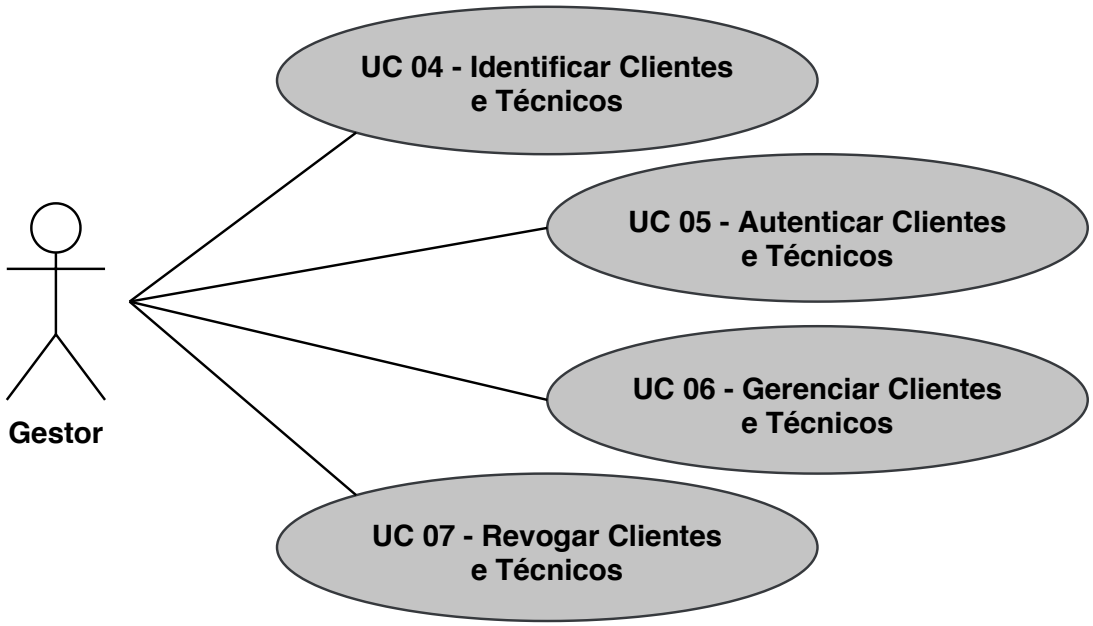
Identificação do Caso de Uso: UC 03 - Autenticar Cliente
Ator Principal: Técnico
Pré-condições: O usuário deve estar logado como "Técnico".
Requisitos Funcionais
RF 03. O software deve permitir que o técnico autentique um cliente.
Requisitos Não Funcionais
RNF 04. O técnico deve ser capaz de operar o sistema a partir de um *smartphone*.
RNF 05. O técnico deve ser capaz de autenticar um cliente via tecnologia *QR Code* quando utilizar um dispositivo móvel.

Fonte: o autor.

4.4.4.3 Gestor

O gestor é o indivíduo responsável por manter (incluir, alterar, visualizar e remover) as informações de clientes e técnicos. Resumidamente, o gestor é a autoridade majoritária da árvore: cliente - técnico e gestor do ISP. Ele é responsável por identificar os clientes e técnicos. Além disso, o gestor também poderá autenticá-los. As funcionalidades do gestor são ilustradas na Figura 6, pelos UC 04, 05, 06 e 07, os quais são descritos, respectivamente, nas Tabelas 6, 7, 8 e 9.

Figura 6 – Casos de Uso do Gestor.



Fonte: o autor.

Tabela 6 – Caso de Uso 04 - Identificar Clientes e Técnicos.

Identificação do Caso de Uso: UC 04 - Identificar Clientes e Técnicos
Ator Principal: Gestor
Pré-condições: O usuário deve estar logado como "Gestor".
Requisitos Funcionais
RF 04. O software deve permitir que o gestor identifique clientes e técnicos.
Requisitos Não Funcionais
RNF 06. O gestor deve ser capaz de realizar todas as operações no sistema a partir de um *smartphone*.

Fonte: o autor.

Tabela 7 – Caso de Uso 05 - Autenticar Clientes e Técnicos.

Identificação do Caso de Uso: UC 05 - Autenticar Clientes e Técnicos
Ator Principal: Gestor
Pré-condições: O usuário deve estar logado como "Gestor".
Requisitos Funcionais
RF 05. O software deve permitir que o gestor autentique clientes e técnicos.
Requisitos Não Funcionais
RNF 06. O gestor deve ser capaz de operar o sistema a partir de um *smartphone*.
RNF 07. O gestor deve ser capaz de autenticar clientes e técnicos via tecnologia *QR Code* quando utilizar um dispositivo móvel.

Fonte: o autor.

Tabela 8 – Caso de Uso 06 - Gerenciar Clientes e Técnicos.

Identificação do Caso de Uso: UC 06 - Gerenciar Clientes e Técnicos
Ator Principal: Gestor
Pré-condições: O usuário deve estar logado como "Gestor".
Requisitos Funcionais
RF 06. O gestor deve ser capaz de adicionar novos clientes e técnicos.
RF 07. O gestor deve ser capaz de alterar informações de clientes e técnicos.
RF 08. O gestor deve ser capaz de visualizar informações de clientes e técnicos.
RF 09. O gestor deve ser capaz de excluir clientes e técnicos.
Requisitos Não Funcionais
RNF 06. O gestor deve ser capaz de realizar todas as operações no sistema a partir de um *smartphone*.

Fonte: o autor.

Tabela 9 – Caso de Uso 07 - Revogar Clientes e Técnicos.

Identificação do Caso de Uso: UC 06 - Gerenciar Clientes e Técnicos
Ator Principal: Gestor
Pré-condições: 1. O usuário deve estar logado como "Gestor".
Requisitos Funcionais
RF 10. O gestor deve ser capaz de revogar a identificação de clientes e técnicos.
Requisitos Não Funcionais
RNF 06. O gestor deve ser capaz de realizar todas as operações no sistema a partir de um *smartphone*.

Fonte: o autor.

5 AVALIAÇÃO

Neste capítulo, apresenta-se uma avaliação do sistema proposto. A avaliação é composta por duas etapas. A primeira consiste em uma avaliação conceitual do protocolo proposto, com a ferramenta Scyther. A segunda etapa demonstra a viabilidade técnica.

5.1 Avaliação Conceitual

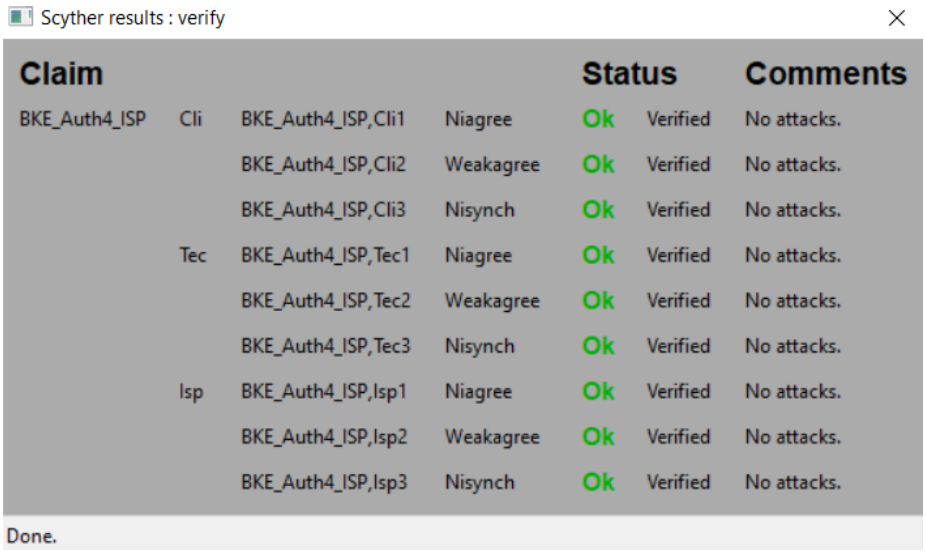
Como primeiro passo para validar conceitualmente os mecanismos propostos, avaliou-se o protocolo BKE-Auth4ISP. Para tanto, utilizou-se a ferramenta Scyther (Cremers, 2006b) em sua versão *Standard* mais recente (1.1.3) para o sistema operacional Microsoft Windows 10 Home Single Language. As configurações são as seguintes: --auto-claims (que gera automaticamente os testes a serem avaliados) e --max-runs= 6 (que determina o número de rodadas realizadas) (Cremers, 2008). A implementação é apresentada no Algoritmo 1 e pode ser encontrada em repositório digital (Quincozes, 2020), para facilitar a reprodutibilidade dos resultados. Há três papéis definidos, cada um em um bloco: Cliente (linhas 4-10), Técnico (linhas 11-18) e Gestor do ISP (linhas 19-25). Para cada mensagem enviada (*e.g.,* linha 7) há uma mensagem recebida correspondente (linha 14).

Algoritmo 1: BKE4ISP na linguagem Scyther.

```
1  usertype String, MessageKey; const ISP-CLI, ISP-TEC, palavraVazia: String;
2  secret otac: MessageKey;
3  protocol BKE-Auth4-ISP(Cli, Tec, Isp){
4  role Cli {
5     var nonce-tec, nonce-isp : Nonce;
6     fresh nonce-cli : Nonce;
7     send_1(Cli, Tec, {nonce-cli, Cli, Isp}otac);
8     recv_3(Isp, Cli, {nonce-cli, nonce-tec, nonce-isp, Tec, Isp}otac);
9     send_4(Cli, Tec, ({nonce-isp}nonce-tec));
10 }
11 role Tec {
12    var nonce-cli, nonce-isp : Nonce;
13    fresh nonce-tec : Nonce;
14    recv_1(Cli, Tec, {nonce-cli, Cli, Isp}otac);
15    send_2(Tec, Isp, {nonce-cli, nonce-tec, Tec, Cli}otac);
16    recv_4(Cli, Tec, ({nonce-isp,nonce-tec}otac));
17    send_5(Tec, Isp, ({nonce-isp}otac));
18 }
19 role Isp {
20    var nonce-tec, nonce-cli : Nonce;
21    fresh nonce-isp : Nonce;
22    recv_2(Tec, Isp, {nonce-cli, nonce-tec, Tec, Cli}otac);
23    send_3(Isp, Cli, {nonce-cli, nonce-tec, nonce-isp, Tec, Isp}otac);
24    recv_5(Tec, Isp, ({nonce-isp}otac));
25 } }
```

As Figuras 7 e 8 apresentam relatórios que apontam os resultados da verificação do protocolo proposto com a ferramenta Scyther.

Figura 7 – Propriedades Niagree, Weakgree, Nisynch.

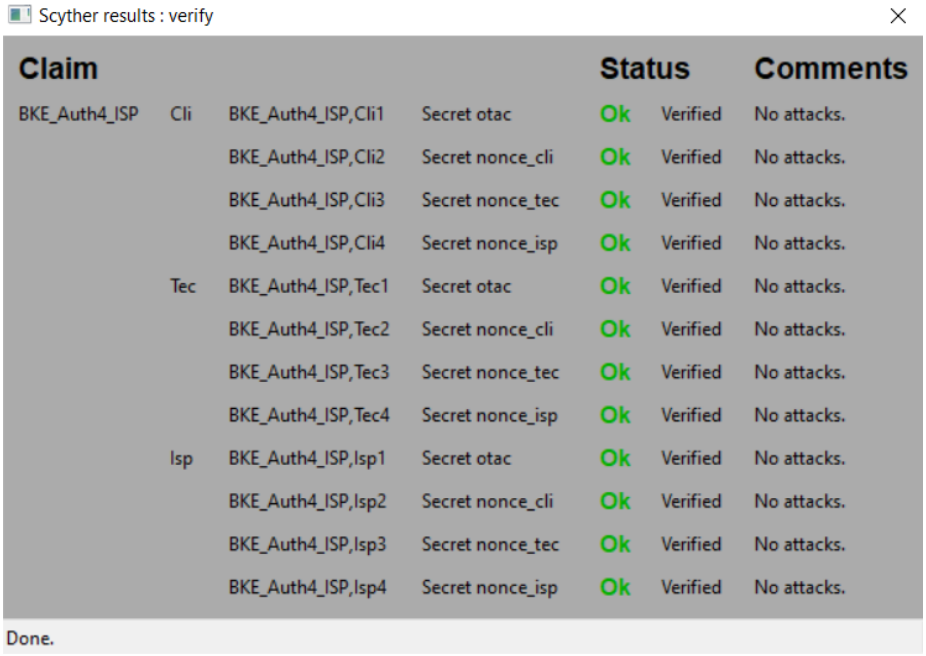

Scyther results : verify

Claim				Status		Comments
BKE_Auth4_ISP	Cli	BKE_Auth4_ISP,Cli1	Niagree	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Cli2	Weakagree	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Cli3	Nisynch	Ok	Verified	No attacks.
	Tec	BKE_Auth4_ISP,Tec1	Niagree	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Tec2	Weakagree	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Tec3	Nisynch	Ok	Verified	No attacks.
	Isp	BKE_Auth4_ISP,Isp1	Niagree	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Isp2	Weakagree	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Isp3	Nisynch	Ok	Verified	No attacks.

Done.

Fonte: o autor.

Figura 8 – Propriedades Secretas.


Scyther results : verify

Claim				Status		Comments
BKE_Auth4_ISP	Cli	BKE_Auth4_ISP,Cli1	Secret otac	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Cli2	Secret nonce_cli	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Cli3	Secret nonce_tec	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Cli4	Secret nonce_isp	Ok	Verified	No attacks.
	Tec	BKE_Auth4_ISP,Tec1	Secret otac	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Tec2	Secret nonce_cli	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Tec3	Secret nonce_tec	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Tec4	Secret nonce_isp	Ok	Verified	No attacks.
	Isp	BKE_Auth4_ISP,Isp1	Secret otac	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Isp2	Secret nonce_cli	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Isp3	Secret nonce_tec	Ok	Verified	No attacks.
		BKE_Auth4_ISP,Isp4	Secret nonce_isp	Ok	Verified	No attacks.

Done.

Fonte: o autor.

A coluna ***Claim*** apresenta o protocolo testado (*BKE_Auth4_ISP*), os indicadores analisados (*e.g., Cli, Tec e Isp*), o indicador único para cada evento (*e.g., BKE_Auth4_ISP, Cli1*) e um evento de afirmação (*e.g., Secret nonce_cli*). Nas colunas ***Status*** e ***Comments*** são reportados possíveis ataques (Jenuario et al., 2020). Observe-se que, para todas as linhas resultantes, o campo *Status* mostra *OK Verified* e que o campo *Comments* apresenta *No Attacks*, o que indica que o protocolo é seguro segundo o método adotado.

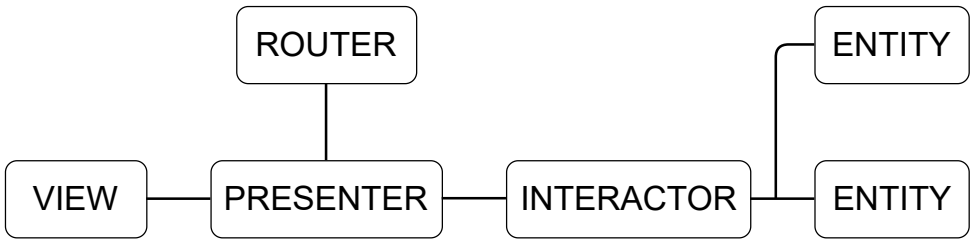
5.2 Demonstração da Viabilidade Técnica

Esta seção demostra a viabilidade técnica da solução proposta, através de um protótipo (Subseção 5.2.1). Tal protótipo é composto por uma aplicação móvel funcional (Subseção 5.2.1.1) integrada com o protocolo de segurança (Subsubseção 5.2.1.2).

5.2.1 Visão Geral do Protótipo

Para demonstrar a viabilidade técnica, implentou-se um protótipo de uma aplicação móvel. No projeto do protótipo, utilizou-se a abordagem *View-Interactor-Presenter-Entity-Router* (VIPER) (Figura 9). Tal abordagem proporciona melhor visualização e organização de pacotes e códigos fontes.

Figura 9 – Arquitetura VIPER.



Fonte: Adaptado de (Android Dev BR, 2019).

Basicamente, o primeiro pacote, denominado *View*, é responsável por caputrar entradas do usuário, exibir informações e redirecionar para o *Presenter*. O *presenter* é responsável por processar requisições, atualizar a *view*, e contatar o *Interactor* e o *Router*. O *Router* tem como objetivo realizar a troca de telas. O *Interactor* é responsável por se conectar com a fonte de dados, no caso deste trabalho, o *Firebase Realtime Database*. Essa camada utiliza a *Entity* para montar resposta de requisições. A última camada, *Entity*, contém a estrutura dos dados utilizados na aplicação.

O aplicativo móvel foi desenvolvido para o sistema operacional Android, utilizando a linguagem de programação *Java*, no ambiente de desenvolvimento Android Studio[1]. Também, para esboçar as interfaces gráficas (Subsubseção 5.2.1.1), utilizou-se o *Extensible Markup Language* (XML). No protótipo, adotou-se a biblioteca Zxing[2] para iniciar o processo de autenticação. A biblioteca permite a geração de *QR Codes* e que um dispositivo Android com hardware de imagem leia códigos de barras e recupere informações. Adotou-se o banco de dados em tempo real *Firebase Realtime Database*[3] para armazenar e recuperar dados durante as interações com a aplicação móvel.

[1] https://developer.android.com/
[2] https://github.com/zxing/zxing
[3] https://firebase.google.com/docs/database

Para gerar funções *hashs*, utilizou-se a biblioteca $MessageDigest$ com a instância $SHA-256$. Ademais, para cifrar e decifrar mensagens, adotou-se o algoritmo *Tiny Encryption Algorithm* (TEA) (Wheeler and Needham, 1994). O algoritmo utiliza tamanho de chaves de 128 bits e criptografia de blocos de 64 bits, podendo ser divididos em blocos de 32 bits (Surendran et al., 2018).

Umas das limitações da presente monografia, é a implementação de apenas um fluxo de autenticação, a saber: técnico solicita autenticação para cliente. Também, não avaliou-se o desempenho da segurança, por fugir do escopo deste trabalho. Outra limitação consiste em assumir que a chave OTAC já está gerada e pronta para uso. Por fim, não implementou-se um fluxo que permita a solicitação de atendimento técnico por um cliente, e, consequentemente, o fluxo de resposta de um gestor designando um técnico para tal atendimento (Quincozes et al., 2020b).

5.2.1.1 Interfaces de Usuário

Nesta seção, as telas de usuário da aplicação móvel são exemplificadas e relacionadas com os respectivos casos de uso, apresentados na Subseção 4.4.4.

- **Telas de Usuário independentes.** Telas independentes são interfaces de usuário que foram implementadas sem estarem previstas na análise de requisitos apresentada na Seção 4.4. Primeiramente, foi necessário elaborar uma tela de login (Figura 10 (a)) para obter controle de acesso e diferenciar privilégios para os Clientes, Técnicos e Gestores. Basicamente, essa tela é composta por dois campos de texto: usuário e senha. Assim que o usuário preenche tais dados e clica no botão **entrar**, o sistema realiza duas operações: (*i*) uma verificação das informações salvas no *firebase realtime database*, e (*ii*) uma validação dos dados para identificar qual é o tipo de usuário (Cliente, Técnico ou Gestor). Caso os dados não sejam válidos, uma mensagem de erro é exibida, conforme ilustrado na Figura 10 (b).

Figura 10 – Telas de login.

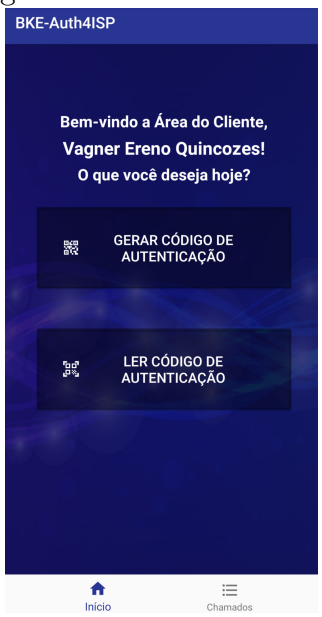
(a) Tela de Login. (b) Credenciais incorretas.

Fonte: o autor.

Após realizar a validação dos dados de login, existem três tipos telas iniciais que podem ser exibidas, dependendo do tipo de usuário: (*i*) Área do Cliente, (*ii*) Área do Técnico ou (*iii*) Área do Gestor. Tais telas e suas respectivas funcionalidades são apresentadas na sequência.

- **Área do Cliente.** A primeira tela, denominada Área do Cliente (Figura 11), é composta por dois menus, ou *fragments*: Início e Chamados. Tais menus são responsáveis por proporcionar ao cliente o acesso às funcionalidades do UC 01 - Autenticar Técnico e do UC 02 - Avaliar Técnico (Subseção 4.4.4.1).

Figura 11 – Área do Cliente.



Fonte: o autor.

O objetivo do primeiro menu, Início, é cumprir o UC 01. Desse modo, existem dois botões: **gerar código de autenticação** e **ler código de autenticação**. O primeiro botão é responsável por gerar um *QR Code* com as informações do técnico (Figura 12). O *QR Code* pode ser utilizado para iniciar o processo de autenticação.

Figura 12 – Gerador de QR Code.

Fonte: o autor.

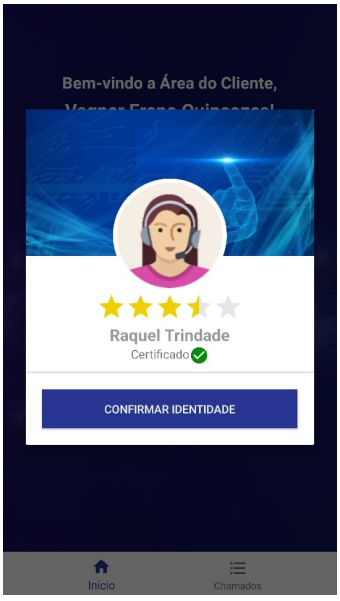
O segundo botão, **ler código de autenticação**, redireciona o cliente para um leitor de código de barras (Figura 13). Tal leitor pode ser utilizado para realizar a leitura e decodificação de *QR Codes*, dando continuidade no processo de autenticação.

Figura 13 – Leitor de Código de Barras.

Fonte: o autor.

Caso o cliente e o técnico forem autenticados com sucesso pelo gestor do ISP, uma interface gráfica para confirmar a identificação do técnico é exibida para o cliente, conforme ilustrado na Figura 14.

Figura 14 – Confirmação de autenticação.

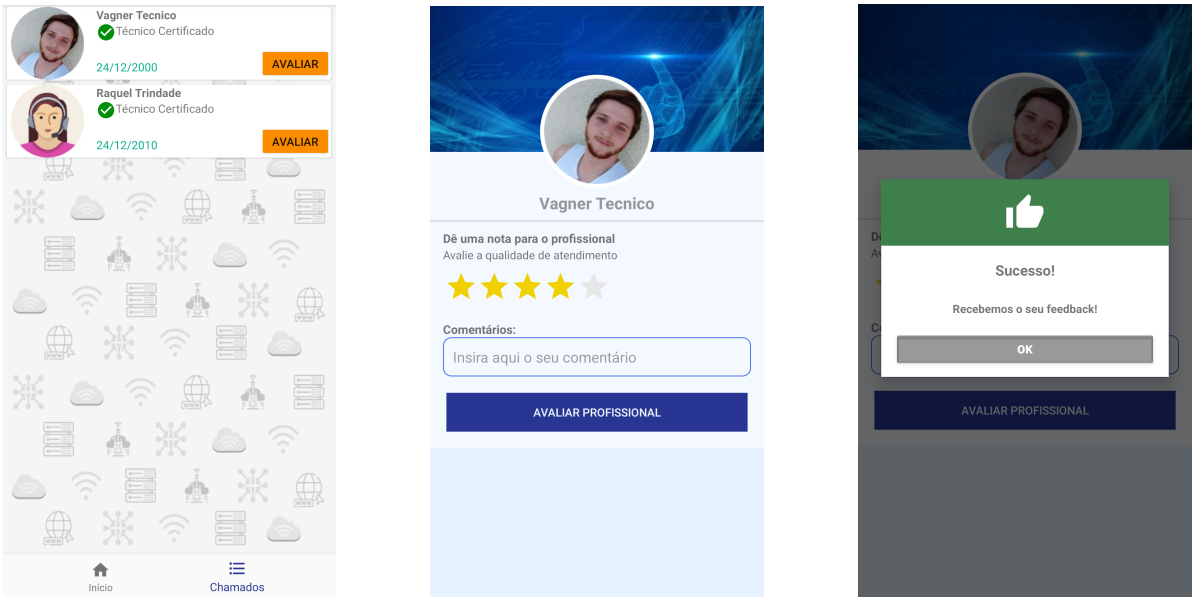


Fonte: o autor.

Antes de confirmar a identidade do técnico, o cliente deve verificar algumas informações, como: foto, nome e certificação. Após a verificação, basta clicar no botão de **confirmar identidades**.

O segundo menu da Figura 11, denominado Chamados, tem como objetivo cumprir o UC 02 (Subseção 4.4.4.1). Assim, as funcionalidades permitem que um cliente avalie um técnico. O processo de avaliação é exemplificado na Figura 15.

Figura 15 – Fluxo para avaliar um técnico.



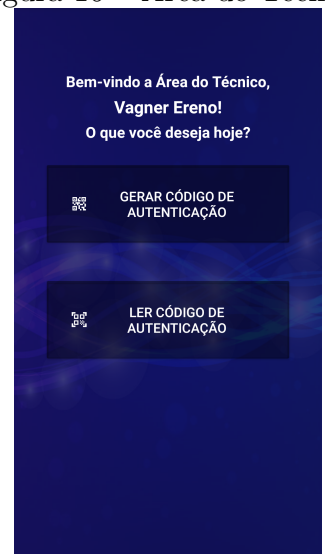
(a) Histórico de chamados. (b) Tela de avaliação. (c) Mensagem de sucesso.

Fonte: o autor.

A Figura 15 (a) mantém um histórico contendo todos os chamados de suporte técnico de um cliente, resumindo a data de atendimento, o nome e a certificação do técnico que realizou o atendimento. Também, existe um botão denominado **avaliar** (representado pela cor laranja). No momento em que o cliente clica neste botão, a tela da Figura 15 (b) é exibida. O cliente deve preencher as informações com a sua avaliação do atendimento técnico, informando: quantidade de estrelas (de uma até cinco), e, opcionalmente, preencher o campo comentário com um relato. Para finalizar o processo, basta clicar no botão **avaliar profissional**. Assim que a avaliação for salva, uma mensagem de sucesso é exibida (Figura 15 (c)).

- **Área do Técnico.** A segunda tela (Área do Técnico, Figura 16), é exibida para indivíduos que pertencem ao tipo de usuário Técnico. Essa tela foi elaborada com o objetivo de cumprir o UC 03 - Autenticar Clientes (Subseção 4.4.4.2). De maneira semelhante a tela exposta na Área do Cliente, existem dois principais botões: **gerar código de autenticação** e **ler código de autenticação**.

Figura 16 – Área do Técnico.



Fonte: o autor.

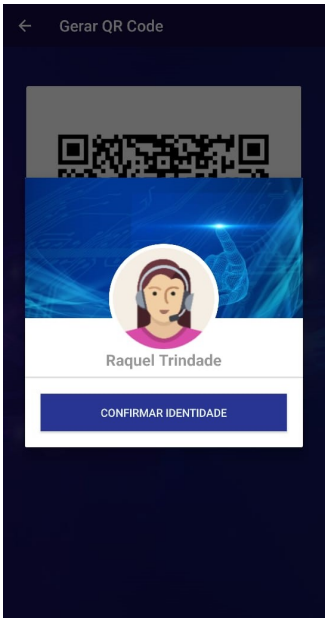
As funcionalidades dos botões também são similares àquelas expostas na Área do Cliente. Ao clicar no botão **gerar código de autenticação**, um *QR Code* com as informações do técnico é gerado (Figura 17 (a)). Ao clicar em **ler código de autenticação**, um leitor de código de barras é exibido (Figura 17 (b)).

Figura 17 – Gerador e Leitor de Códigos de Autenticação.

(a) Gerar código de autent. (b) Ler código de autent.

Por fim, a Figura 18 exemplifica a interface gráfica exibida ao fim do processo de autenticação, no momento em que o gestor do ISP verifica ambas identidades (do cliente e do técnico) e autoriza a autenticação. O técnico deve verificar as informações de identificação do cliente, como: foto e nome. Após a verificação, para confirmar a identificação do cliente basta clicar em **confirmar identidade**.

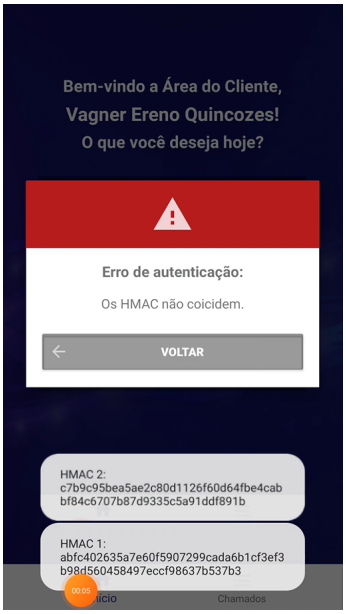
Figura 18 – Confirmação de autenticação.



Fonte: o autor.

Se caso houver erro durante o processo de autenticação, por exemplo, os HMAC das mensagens não coincidam, a mensagem da Figura 19 é exibida.

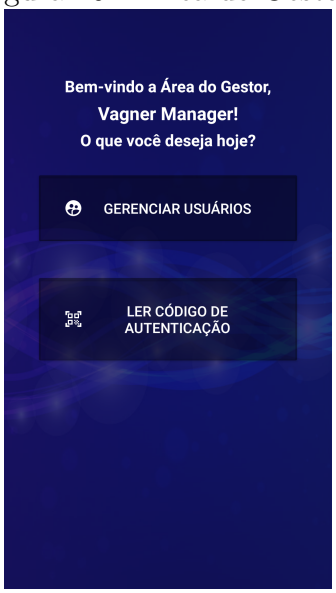
Figura 19 – Mensagem de erro: HMAC não coincide.



Fonte: o autor.

- **Área do Gestor.** Nessa área, os seguintes UCs são cumpridos: 04, 05, 06 e 07 (Subseção 4.4.4.3). Ao entrar na área do gestor, dois botões são visíveis: (*i*) gerenciar usuários e (*ii*) ler código de autenticação (Figura 20).

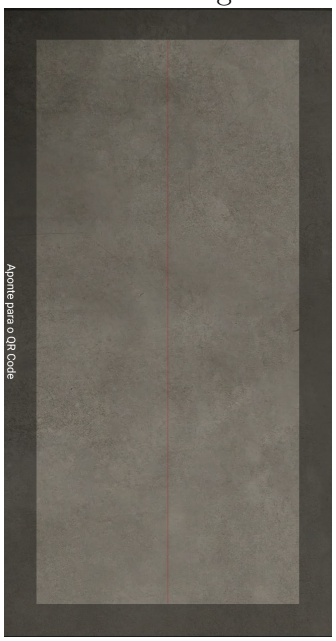
Figura 20 – Área do Gestor.



Fonte: o autor.

De maneira semelhante às áreas do cliente e do técnico, ao clicar no botão **ler código de autenticação**, o gestor é direcionado para um leitor de código de barras (Figura 21). Desse modo, ao utilizar o leitor de código de barras, o gestor pode realizar autenticação com outro usuário do sistema (UC 05).

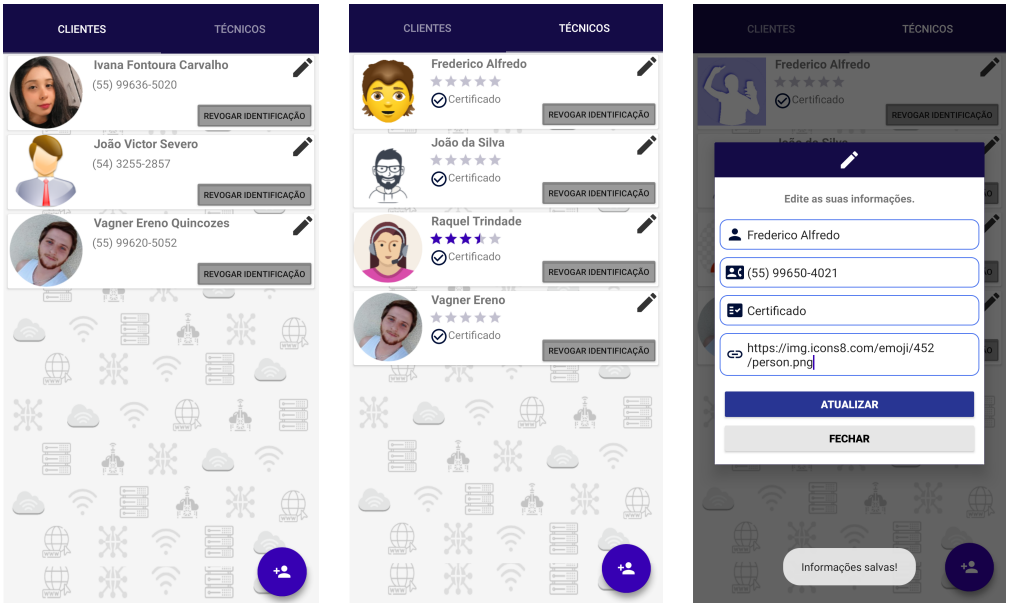
Figura 21 – Leitor de Código de Autenticação.



Fonte: o autor.

Na sequência, ao clicar no botão **gerenciar usuários**, o gestor é redirecionado para a próxima tela. Essa tela contém funcionalidades específicas para cumprir os UC 06 - Gerenciar usuários, UC 04 - Identificar usuários e UC 07 - Revogar identificação. Primeiramente, a tela é dividida por duas abas: (*i*) clientes, e (*ii*) técnicos (Figura 22 (a) e (b)). Ao clicar na aba **clientes**, uma lista contendo *cards* de todos os clientes cadastrados no ISP é exibida. Os *cards* são compostos por algumas informações, como: foto, nome e telefone. Ademais, ao clicar na aba de **técnicos**, uma lista com *cards* de todos os técnicos cadastrados no ISP é exibida. As informações que compõem o *card* são: foto, nome, avaliação e certificação.

Figura 22 – Lista de clientes e técnicos.



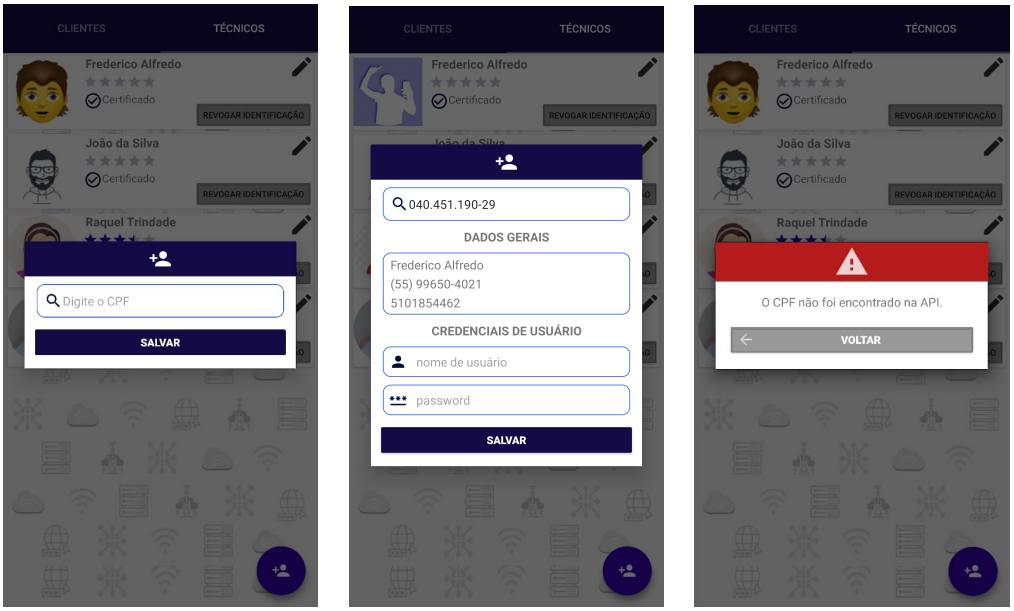
(a) Clientes. (b) Técnicos. (c) Editar informações.

Fonte: o autor.

As funcionalidades expostas acima compõem o UC 06, proporcionando ao gestor uma lista contendo os dados de todos os usuários cadastrados no ISP. Além disso, para complementar o UC 06, existe a possibilidade de editar os dados de usuários. Para tanto, basta clicar no botão de editar (representado por um lápis, no canto direito do *card* de cada usuário). Ao clicar nesse botão, uma tela de editar é exibida (Figura 22 (c)). A tela é composta por informações pré-preenchidas do usuário, como: nome, telefone, certificação e foto. Para realizar a edição, basta substituir os dados pré-preenchidos pelas novas informações e clicar no botão **atualizar**. A partir do momento em que as informações são alteradas, uma mensagem *toast* de sucesso é exibida.

Resumidamente, as interfaces gráficas ilustradas na Figura 23 têm três principais objetivos: (*i*) cumprir o UC 04, permitindo que o gestor do ISP adicione e identifique novos usuários, (*ii*) definir usuário e senha para que clientes e técnicos possam acessar a aplicação móvel, e (*iii*) simular a integração dos dados com uma API externa.

Figura 23 – Identificar clientes e técnicos.



(a) Campo para digitar CPF. (b) Busca por CPF. (c) CPF não encontrado.

Fonte: o autor.

Para adicionar e identificar novos usuários no sistema, simula-se o funcionamento de uma API externa. Também, assume-se que as informações de novos usuários surgem a partir dela. Nesse sentido, basta clicar no botão de adicionar (representado por um ícone de uma pessoa e um +, no canto inferior direito da tela) e preencher o campo com o CPF do usuário, conforme ilustrado na Figura 23 (a). A partir desse momento, é realizada uma busca na API externa pelo CPF digitado. Se o CPF estiver cadastrado na API, automaticamente as informações nome, telefone e RG são exibidas na tela (Figura 23 (b)). Os campos de credenciais de usuário servem para que o gestor possa definir um nome de usuário e uma senha de acesso ao aplicativo para o novo usuário. Se o CPF não for encontrado na API, a mensagem de erro ilustrada na Figura 23 (c) é apresentada ao gestor.

Por fim, a interface gráfica referente ao UC 07 é exemplificada na Figura 24. A partir do momento em que o gestor do ISP clica no botão **revogar identificação**, localizado na parte inferior direita do *card* de cada usuário, a chave OTAC (utilizada para cifrar e decifrar mensagens durante o processo de autenticação) armazenada no *firebase realtime database* é substituida por uma *string* "**REVOGADA**".

Figura 24 – Revogar Identificação.

Fonte: o autor.

Neste contexto, considerando que a chave OTAC de tal usuário não é mais válida, não será possível realizar autenticação com os demais usuários do sistema, por inconsistência de chave. A Figura 25 ilustra a mensagem de erro exibida ao usuário.

Figura 25 – Erro de autenticação.

Fonte: o autor.

5.2.1.2 Implementação do Protocolo de Segurança

Uma implementação do protocolo BKE-Auth4ISP foi realizada como prova de conceito. Além disso, o protocolo está integrado com a aplicação móvel explanada nas seções anteriores. Na implementação, considerou-se a versão do protocolo BKE-Auth4ISP exposta na Tabela 10. A versão implementada inicia com o técnico solicitando autenticação. Assim, o técnico gera o código de autenticação e o cliente lê o código de autenticação.

Tabela 10 – Protocolo BKE-Auth4ISP: Técnico solicita autenticação.

M1	Técnico → Cliente	[E(Tec, Isp, nonce-tec)]HMAC
M2	Cliente → Gestor do ISP	[E(Cli, Tec, nonce-tec, nonce-cli)]HMAC
M3	Gestor do ISP → Técnico	[E(nonce-tec, nonce-cli, nonce-isp, Cli, Isp)]HMAC
M4	Técnico → Cliente	[E(nonce-isp,nonce-cli)]HMAC
M5	Cliente → Gestor do ISP	[E(nonce-isp)]HMAC

Fonte: o autor.

Na implementação, assume-se que o OTAC é definido no momento em que o gestor atribui um técnico para realizar um atendimento de suporte a um cliente. Também, para realizar autenticação assume-se que a entidade Gestor do ISP é representada por um servidor automatizado, implementado na linguagem de programação **java** com conexão ao ***firebase realtime database***. Basicamente, o gestor do ISP aguarda uma mensagem contendo as informações das entidades. Assim que a mensagem chega, ele realiza uma verificação e envia uma resposta confirmando ou não a autenticação.

Em geral, as mensagens da Tabela 10 são compostas por suas respectivas informações e um HMAC das mesmas. As mensagens são cifradas pelo algoritmo TEA, cuja chave secreta é um OTAC. A seguir, são ilustradas as saídas das mensagens (M) implementadas. Vale observar que, a identificação do técnico (Tec) é representada por *vequincozes*, a identificação do cliente (Cli) é representada por *vequincozes*2, e a identificação do gestor do ISP (Isp) é representada por $INTERNEITH$. Os *nonces* são números aleatórios.

O processo de autenticação (Figura 26) inicia quando o técnico clica no botão gerar código de autenticação (Figura 17 (a)). Nesse momento, uma mensagem contendo a sua identificação (`Tec`), a identificação do ISP (`Isp`) e um *nonce* de autenticação (`nonce-tec`) é formada (*Input*, linha 1). Na sequência, um HMAC dessas informações é gerado (linha 2). A mensagem é cifrada com o algoritmo TEA, cuja chave é o OTAC. Por fim, um *QR Code* é gerado através das informações cifradas mais o HMAC (linha 3).

Figura 26 – M1: Técnico → Cliente [E(Tec, Isp, nonce-tec)]HMAC.

```
1: Input:  vequincozes=INTERNEITH=1235600746547134540
2: HMAC: f91f42c739c77775e829beecd92d97bb806f6ee4bff22ab2112b51805d2d7cf6
3: QR Code = Encrypted Input + HMAC:
 AAAAa2g6cQwr2yuPSieh3t5wAO5xx7q80niXVpU5OZ38im4vHdGOzcv7JihBG2YIOK0yv3leEFX7mN
 A1IJuEXaahg289OoPrcYTFEUaQA9PJZbtYheN9UO0aXfgJ08FTb5ShxNQlaM76tXKxJUCFrcy+z6s=
```

A próxima etapa requer que o cliente clique no botão de ler código de autenticação (Figura 13). O processo está ilustrado na Figura 27. Após abrir o leitor de código de autenticação (Figura 13), o cliente deve redirecionar a câmera do seu *smartphone* para o *QR Code* gerado pelo técnico. Assim que o *QR Code* é detectado, o cliente utiliza o algoritmo TEA com a sua chave OTAC para decifrá-lo (linha 1). Após decifrar, o cliente verifica se o HMAC que consta na mensagem do técnico coincide com o HMAC computado das informações recebidas (linha 2). Se coincidir, o cliente adiciona na mensagem a sua identificação (`Cli`), um *nonce* de autenticação (`nonce-cli`) e um `HMAC` dessas informações (linha 3). Esta mensagem é cifrada com o algoritmo TEA cuja chave é o OTAC (linha 4) e enviada para o gestor do ISP.

Figura 27 – M2: Cliente → Gestor [E(Cli, Tec, nonce-tec, nonce-cli)]HMAC.

```
1: M1 Decrypted:  vequincozes=INTERNEITH=1235600746547134540
 =f91f42c739c77775e829beecd92d97bb806f6ee4bff22ab2112b51805d2d7cf6
2: Validação de HMAC: HMAC Output == HMAC Result
   HMAC Output:  f91f42c739c77775e829beecd92d97bb806f6ee4bff22ab2112b51805d2d7cf6
   HMAC Result:  f91f42c739c77775e829beecd92d97bb806f6ee4bff22ab2112b51805d2d7cf6
3: M2 Input:  vequincozes2=vequincozes=1235600746547134540=5529155482608127700
 =c85e31093447435c99768f75e583381025e63f495992a699487a172fce7eeab0
4: M2 Encrypted:  AAAAgWg6cQwr2yuPCPg5E/xfBe6kn+7kV0nB4i4pYIllW2Lb4+n
 /z3GSV0FdfSlyQw7PyiwSsws+VH6W
 +br7RostpmGHGEL5ZlOj5CReHHYDrHzg1HORcV2ZUChVfPIpXP34GTEKW7/tLE/19
 /3BKRJISP44OA37grh1VejnHt9s9wUFN8PWM+CSCHk=
```

Fonte: o autor.

O Gestor do ISP recebe a solitação de autenticação, decifra a mensagem utilizando o algoritmo TEA com a chave OTAC (Figura 28, linha 1) e computa os HMAC (linha 2). Se os HMAC coicidirem, o gestor gera uma mensagem contendo os nonces de autenticação do técnico (`nonce-tec`), do cliente (`nonce-cli`), do ISP (`nonce-isp`), mais a identificação do cliente (`Cli`), a sua identificação (`Isp`) e um (`HMAC`) dessas informações (linha 3). Essa mensagem é cifrada (linha 4) com o algoritmo TEA cuja chave é um OTAC e enviada para o técnico.

Figura 28 – M3: Gestor → Técnico [E(nonce-tec, nonce-cli, nonce-isp, Cli, Isp)]HMAC.

```
1: M2 Decrypted:  vequincozes2=vequincozes=1235600746547134540=552915548260812770
   0=c85e31093447435c99768f75e583381025e63f495992a699487a172fce7eeab0
2: Validação de HMAC: HMAC Output == HMAC Result
   HMAC Output:  c85e31093447435c99768f75e583381025e63f495992a699487a172fce7eeab0
   HMAC Result:  c85e31093447435c99768f75e583381025e63f495992a699487a172fce7eeab0
3: M3 Input:  5529155482608127700=1235600746547134540=5772564424859518473=vequinc
   ozes2=INTERNEITH=423933df2530b2fc25a44e5d6699cfaceb701e24c462d0e1d5f9ae934be0
   d777
4: M3 Encrypted:  AAAAlA8+qXfswOjphJoRwqEztns/mf+5yuGWEghbniPkDw7jXdlqqxQedArvBJ2
   Y3/yiEsur6B5sY5czccaYL7Dr8lPfmo8drSXCw0AVtFqXAcYCKFp8IqC5LOPQ6l/EbRO6q/k/63sW
   /lQPKIoHZktDcJPrvRrStqkaHwh+HoyiDv7qLkIzT7HQbdl+hg64oewj8kfPxTIB44m7
```

Fonte: o autor.

Ao receber a mensagem do gestor do ISP, o técnico decifra a mensagem utilizando o algoritmo TEA com a chave OTAC (Figura 29, linha 1), confere os HMACs (linha 2), gera a próxima mensagem contendo os nonces de autenticação (`nonce-isp`) e (`nonce-cli`), mais um (`HMAC`) das informações (linha 3), cifra a mensagem utilizando o algoritmo TEA cuja chave é um OTAC (linha 4) e envia para o cliente.

Figura 29 – M4: Técnico → Cliente [E(nonce-isp,nonce-cli)]HMAC.

```
1: M3 Decrypted:  5529155482608127700=1235600746547134540=5772564424859518473
 =vequincozes2=INTERNEITH
 =423933df2530b2fc25a44e5d6699cfaceb781e24c462d8e1d5f9ae934be0d777
2: Validação de HMAC: HMAC Output == HMAC Result
   HMAC Output:  423933df2530b2fc25a44e5d6699cfaceb781e24c462d8e1d5f9ae934be0d777
   HMAC Result:  423933df2530b2fc25a44e5d6699cfaceb781e24c462d8e1d5f9ae934be0d777
3: M4 Input:  5772564424859518473=1235600746547134540
4: M4 Encrypted:  AAAAaO8EnZjf/KISy6voHmxjlzN
 /huhg5oSFpQhbniPkDw7jXdlqqxQedApxkSIjeTvRCFTiBPdhj7uF8flzGtli6TzcudpdGCidG6bO4
 aX19U5RS0YAcaA1mGnsPmrRugFWO2ajNsuWqdy1
```

Fonte: o autor.

Ao receber a mensagem do técnico, o cliente decifra a mensagem utilizando o algoritmo TEA cuja chave é o OTAC (Figura 30, linha 1), verifica se os HMAC conferem (linha 2), e envia a última mensagem cifrada (linha 4) para o gestor do ISP contendo o *nonce* de autenticação (`nonce-isp`) (linha 3).

Figura 30 – M5: Cliente → Gestor [E(nonce-isp)]HMAC.

```
1: M4 Decrypted:  5772564424859518473=1235600746547134540
 =3577fb7c9f2eff024ff8f8a6eed7678f3445fc7c17e70790f71361532600122a
2: Validação de HMAC: HMAC Output == HMAC Result
   HMAC Output:  3577fb7c9f2eff024ff8f8a6eed7678f3445fc7c17e70790f71361532600122a
   HMAC Result:  3577fb7c9f2eff024ff8f8a6eed7678f3445fc7c17e70790f71361532600122a
3: M5 Input:  5772564424859518473
4: M5 Encrypted:  AAAAVO8EnZjf/KISy6voHmxjlzPfHFQX
 +BntXp08swqWloq0ZrUB1BdxiZRx2wlyZJtyjZwjn1C0/8lVsRNC4
 /5IJVzbqAuU7KVptUcQfdfa54U6fHjfegDrTXQ=
```

Fonte: o autor.

Por fim, a última etapa do processo de autenticação acontece quando o gestor do ISP recebe a M5 do cliente. Ao receber a M5, o gestor decifra a mensagem utilizando o algoritmo TEA com a chave OTAC (linha 1), computa os HMACs (linha 2) e encerra o processo de autenticação (Figura 31).

Figura 31 – Processo de autenticação finalizado.

```
1: M5 Decrypted:    5772564424859518473=8598763723dede820bbedccf55d23
   6dc34e09fd1d8b190d648683e9824f006a2
2: Validação de HMAC: HMAC Output == HMAC Result
   HMAC Output:    8598763723dede820bbedccf55d236dc34e09fd1d8b190d648
   683e9824f006a2
   HMAC Result:    8598763723dede820bbedccf55d236dc34e09fd1d8b190d648
   683e9824f006a2
```

Fonte: o autor.

Vale ressaltar que caso os HMAC não coincidam em alguma das mensagens, a aplicação móvel exibe uma mensagem de erro (Figura 19) e a autenticação não é realizada.

6 CONSIDERAÇÕES FINAIS

Neste capítulo, apresenta-se as conclusões finais, resultados alcançados e os trabalhos futuros.

6.1 Conclusões

Neste trabalho, propõe-se um sistema seguro para realizar autenticação e identificação entre clientes, técnicos e gestores de ISPs. O sistema é composto por (*i*) uma instanciação de um protocolo de autenticação combinado com um protocolo recente, eficiente e seguro para identificação, e (*ii*) uma aplicação móvel que instancia os protocolos propostos de maneira amigável. Os protocolos de segurança foram verificados formalmente utilizando a ferramenta Scyther, que constatou que os protocolos não apresentam falhas. Também, uma análise da viabilidade técnica foi realizada, através da prototipação de um aplicativo móvel para *smartphones* integrado com os protocolos propostos.

6.2 Resultados

Os principais resultados deste trabalho de conclusão de curso são listados abaixo:

- Publicação na Escola Regional de Redes de Computadores (ERRC 2020) (Quincozes et al., 2020a).
- Publicação nos Anais do Salão Internacional de Ensino, Pesquisa e Extensão (SIEPE 2020) (Quincozes et al., 2020b).
- (Em andamento) Registro do protótipo no Instituto Nacional da Propriedade Industrial (INPI).
- (Em andamento) Elaboração de artigo para o Simpósio Brasileiro de Segurança da Informação e Sistemas Computacionais (SBSeg 2021).

6.3 Trabalhos Futuros

Como trabalhos futuros, pretende-se superar as limitações deste trabalho. Especificamente, planeja-se estender e complementar a aplicação móvel. Por exemplo, pretende-se incluir a complementação do fluxo de autenticação e a geração da chave OTAC. Também, pensa-se em realizar avalições complementares do sistema, por exemplo, considerando aspectos de desempenho e segurança do protótipo.

Para além disso, vislumbra-se outras oportunidades de pesquisa, desenvolvimento e inovação. Por exemplo, planeja-se a integração do sistema com o Gateway de Acesso Controlado (GAC) (Torres et al., 2020), que está sendo desenvolvido no âmbito do mesmo projeto de pesquisa e inovação deste trabalho. Também, pensa-se em agregar, pelo menos,

duas novas funcionalidades na aplicação. A primeira constitui-se de uma interface para que o gestor possa designar um determinado técnico para o cliente. A segunda consiste no desenvolvimento de uma camada de inteligência para, por exemplo, distribuição de atendimentos entre técnicos, considerando cenários desafiadores, como no caso de enxurrada de demanda (Quincozes et al., 2020b).

REFERÊNCIAS

S. F. Aghili, H. Mala, P. Kaliyar, and M. Conti. Seclap: Secure and lightweight rfid authentication protocol for medical iot. *Future Generation Computer Systems*, 101: 621–634, 2019. Citado 2 vezes nas páginas 25 e 27.

P. Anand, Y. Singh, A. Selwal, M. Alazab, S. Tanwar, and N. Kumar. Iot vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges. *IEEE Access*, 8:168825–168853, 2020. Citado na página 19.

D. C. Android Dev BR. Arquitetura limpa nas apps: utilizando viper no android, 2019. URL <https://medium.com/android-dev-br/arquitetura-limpa-nas-apps-utilizando-viper-no-android-f39e51b44723>. Citado na página 41.

A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, et al. The avispa tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification*, p. 281–285. Springer, 2005. Citado na página 22.

D. Basin, C. Cremers, and S. Meier. Provably repairing the iso/iec 9798 standard for entity authentication 1. *Journal of Computer Security*, 21(6):817–846, 2013. Citado na página 23.

L. W. d. Bettio. O crescimento da internet no Brasil, serviços e regulamentação. 2016. Citado na página 17.

B. Blanchet. A computationally sound mechanized prover for security protocols. *IEEE Transactions on Dependable and Secure Computing*, 5(4):193–207, 2008. Citado na página 22.

B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre. Proverif 2.00: automatic cryptographic protocol verifier, user manual and tutorial. *Version from*, p. 05–16, 2018. Citado na página 22.

S. Bojjagani and V. Sastry. A secure end-to-end proximity NFC-based mobile payment protocol. *Computer Standards & Interfaces*, 66:103348, 2019. Citado na página 24.

P. Chandrakar, A. Jain, S. Balivada, and R. Ali. A secure authentication protocol for vehicular ad-hoc networks. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, p. 1–7. IEEE, 2019. Citado 2 vezes nas páginas 25 e 27.

Y. Cherdantseva and J. Hilton. A reference model of information assurance & security. In *2013 International Conference on Availability, Reliability and Security*, p. 546–555. IEEE, 2013. Citado na página 20.

C. Cremers. Feasibility of multi-protocol attacks. In *First International Conference on Availability, Reliability and Security (ARES'06)*, p. 8–pp. IEEE, 2006a. Citado na página 23.

C. Cremers. Key exchange in ipsec revisited: Formal analysis of ikev1 and ikev2. In *European Symposium on Research in Computer Security*, p. 315–334. Springer, 2011. Citado na página 23.

C. Cremers and S. Mauw. A family of multi-party authentication protocols. In *First Benelux Workshop on Information and System Security (WISSec)*, 2006. Citado 3 vezes nas páginas 17, 25 e 30.

C. J. Cremers. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In *International conference on computer aided verification*, p. 414–418. Springer, 2008. Citado 2 vezes nas páginas 23 e 39.

C. J. F. Cremers. *Scyther: Semantics and verification of security protocols*. Eindhoven university of Technology Eindhoven, Netherlands, 2006b. Citado 3 vezes nas páginas 18, 22 e 39.

D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983. Citado na página 30.

M. J. Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions. 2015. Citado na página 22.

I. O. for Standardization/International Electrotechnical Commission et al. Iso/iec 18092 information technology—telecommunications and information exchange between systems—near field communication—interface and protocol (nfcip-1). *ISO/IEC*, 18092, 2004. Citado na página 23.

E. Haselsteiner and K. Breitfuß. Security in near field communication (nfc). In *Workshop on RFID security*, p. 12–14. sn, 2006. Citado na página 24.

T. Jenuario, J. O. Chervinski, G. Paz, R. Beltran, R. Fernandes, and D. Kreutz. Verificação Automática dos Protocolos de Segurança Needham-Schroeder, WMF e CSA com a ferramenta Scyther. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, 3(1), 2020. Citado na página 40.

P.-S. Jeong and Y.-H. Cho. Smartphone User Authentication Algorithm based on Mutual Cooperation in Mobile Environment. *Journal of the Korea Institute of Information and Communication Engineering*, 21(7):1393–1400, 2017. Citado 2 vezes nas páginas 25 e 26.

C. R. Junior, S. E. Quincozes, and J. F. Kazienko. LegitimateBroker: Mitigando Ataques de Personificaç ao em Broker MQTT na Internet das Coisas. p. 1–14, 2019. Citado 2 vezes nas páginas 25 e 27.

M. Koschuch, M. Hudler, H. Eigner, and Z. Saffer. Token-based Authentication for Smartphones. In *2013 International Conference on Data Communication Networking (DCNET)*, p. 1–6. IEEE, 2013. Citado 2 vezes nas páginas 25 e 26.

D. Kreutz, J. Yu, F. M. V. Ramos, and P. Esteves-Verissimo. ANCHOR: Logically centralized security for software-defined networks. *ACM Transactions on Privacy and Security*, 22(2):8:1–8:36, 2019. ISSN 2471-2566. doi: <10.1145/3301305>. URL <http://doi.acm.org/10.1145/3301305>. Citado 2 vezes nas páginas 17 e 31.

D. Kreutz, R. Fernandes, G. Paz, T. Jenuario, R. Mansilha, R. Immich, and C. C. Miers. Auth4App: Protocols for Identification and Authentication using Mobile Applications. In *SBC 20th International Brazilian Symposium on Information and Computational Systems Security (SBSeg)*, p. 1–14. SBC, 2020a. Citado 4 vezes nas páginas 17, 25, 27 e 30.

D. Kreutz, R. Mansilha, S. E. Quincozes, T. Jenuario, and J. O. Chervinski. Introducão a verificação automática de protocolos de seguranca com scyther. In *Minicurso na XVIII Escola Regional de Redes de Computadores*, Joinville-SC, Brasil, nov 2020b. SBC. Citado na página 22.

Z. Lei, Y. Nan, Y. Fratantonio, and A. Bianchi. On the insecurity of sms one-time password messages against local attackers in modern mobile devices. p. 1–18, 01 2021. doi: <10.14722/ndss.2021.24212>. Citado 2 vezes nas páginas 25 e 27.

Y. Liang, H. V. Poor, and S. Shamai. *Information theoretic security*. Now Publishers Inc, 2009. Citado na página 20.

T. Marktscheffel, W. Gottschlich, W. Popp, P. Werli, S. D. Fink, A. Bilzhause, and H. de Meer. QR Code Based Mutual Authentication Protocol for Internet of Things. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, p. 1–6. IEEE, 2016. Citado 3 vezes nas páginas 23, 25 e 26.

S. Meier, B. Schmidt, C. Cremers, and D. Basin. The tamarin prover for the symbolic analysis of security protocols. In *International Conference on Computer Aided Verification*, p. 696–701. Springer, 2013. Citado na página 22.

C. Mulliner. Vulnerability analysis and attacks on nfc-enabled mobile phones. In *2009 International Conference on Availability, Reliability and Security*, p. 695–700. IEEE, 2009. Citado na página 23.

D. R. Patel. *Information security: theory and practice*. PHI Learning Pvt. Ltd., 2008. Citado na página 21.

A. Pratama and E. Prima. 2FMA-NetBank: A Proposed Two Factor and Mutual Authentication Scheme for Efficient and Secure Internet Banking. In *2016 8th International Conference (ICITEE)*, p. 1–4. IEEE, 2016. Citado 3 vezes nas páginas 17, 25 e 26.

S. E. Quincozes. Uma arquitetura segura baseada na computação ubíqua para recuperação de registros médicos. 2015. Citado na página 21.

S. E. Quincozes and J. F. Kazienko. A secure architecture based on ubiquitous computing for medical records retrieval. In *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, p. 1–8. IEEE, 2016. Citado na página 24.

V. E. Quincozes. Repositório digital: BKE4ISP. Disponível em: https://github.com/vagnerereno/autenticacaoparaisps.git, 2020. Citado na página 39.

V. E. Quincozes, D. Temp, S. E. Quincozes, D. Kreutz, and R. B. Mansilha. Sistema para Autenticação entre Clientes, Técnicos e ISPs. In *5o Workshop Regional de Segurança da Informação e de Sistemas Computacionais*, Joinville-SC, Brasil, nov 2020a. URL <http://errc.sbc.org.br/2020/wrseg.php>. Citado 3 vezes nas páginas 18, 25 e 57.

V. E. Quincozes, R. B. Torres, R. B. Mansilha, and D. Kreutz. Aplicativo de conexão entre usuários, técnicos e gestores de isps para suportar enxurradas de demandas. *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, 12(2), 2020b. Citado 3 vezes nas páginas 42, 57 e 58.

R. Rivest and S. Dusse. The md5 message-digest algorithm, 1992. Citado na página 22.

R. L. Rivest. The md4 message digest algorithm. In *Conference on the Theory and Application of Cryptography*, p. 303–311. Springer, 1990. Citado na página 22.

A. Savage. Md5 and sha-1 collision attacks: A tutorial. 2008. Citado na página 22.

R. Shirey. Internet security glossary, 2000. Citado na página 21.

W. Stallings. *Cryptography and network security, 4/E*. Pearson Education India, 2006. Citado 3 vezes nas páginas 19, 21 e 22.

W. Stallings, G. Bressan, and A. Barbosa. *Criptografia e segurança de redes* . Pearson Educacion, 2008. Citado 2 vezes nas páginas 20 e 21.

S. Surendran, A. Nassef, and B. D. Beheshti. A survey of Cryptographic Algorithms for IoT Devices. In *2018 IEEE Long Island Systems, Applications and Technology Conference*, p. 1–8, 2018. Citado na página 42.

C. Thammarat. Efficient and secure nfc authentication for mobile payment ensuring fair exchange protocol. *Symmetry*, 12(10):1649, 2020. Citado na página 24.

R. Torres, V. E. Quincozes, R. B. Mansilha, and D. Kreutz. Gateway de acesso controlado-gac. *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, 12(2), 2020. Citado na página 57.

D. Wave. Information technology automatic identification and data capture techniques qr code bar code symbology specification. *International Organization for Standardization, ISO/IEC*, 18004, 2015. Citado na página 23.

D. J. Wheeler and R. M. Needham. Tea, a tiny encryption algorithm. In *International workshop on fast software encryption*, p. 363–366. Springer, 1994. Citado na página 42.

P. Williams, I. Dutta, H. Daoudm, and M. Bayoumi. Security Aspects of Internet of Things – A Survey. In *6th IEEE World Forum on Internet of Things*, p. 1–6, 2020. Citado na página 22.

L. Wu, H. J. Cai, and H. Li. Sgx-uam: A secure unified access management scheme with one time passwords via intel sgx. *IEEE Access*, 9:38029–38042, 2021. doi: <10.1109/ACCESS.2021.3063770>. Citado 2 vezes nas páginas 25 e 27.

T. G. Zimmerman. Personal area networks: near-field intrabody communication. *IBM systems Journal*, 35(3.4):609–617, 1996. Citado na página 23.

ÍNDICE