Uma Proposta de Protocolo Para Liberação de Acesso Seguro a Redes Sem Fio Públicas Gratuitas

Pedro Victor A. Melo[1], **Roberto Samarone Dos Santos Araújo**[1]

[1]Faculdade de Computação – Instituto de Ciências Exatas e Naturais
Universidade Federal do Pará (UFPA) – Belém – PA –Brasil

pedro.melo@icen.ufpa.br, rsa@ufpa.br

Abstract. *Wireless networks have become popular and are widely used to provide internet access. They are used by the brazilian government in programs for the digital inclusion of citizens. These free and public networks generally have few (if any) security mechanisms to identify users. Consequently, users can commit illegal acts without being identified. In this context, this work aims to present a proposal for a protocol for secure connection to public and free wireless networks. It makes it possible to to track users if they commit crimes.*

Resumo. *Redes sem fio tornaram-se populares e são amplamente utilizadas para acessar a internet. Elas são utilizadas pelo governo brasileiro em programas para a inclusão digital dos cidadãos. Essas redes públicas e gratuitas geralmente apresentam poucos (ou nenhum) mecanismos de segurança que possam identificar usuários. Consequentemente, esses usuários podem cometer atos ilícitos sem serem identificados. Nesse contexto, este trabalho tem como objetivo apresentar uma proposta de protocolo para conexão segura a redes sem fio públicas e gratuitas. Ele possibilita identificar os usuários, para possível rastreamento em caso de cometimento de crimes.*

1. Introdução

Redes sem fio são a principal forma de acesso a internet no mundo. Sua popularização se deu justamente por não necessitar de fios para estabelecer a conexão. Isso permitiu assim uma maior mobilidade dos usuários. Um exemplo disso é a rede [NAVEGAPARA 2014], que é a principal forma de inclusão digital utilizada pelo estado do Pará, auxiliando no combate a desigualdade social.

A simples disponibilização desse tipo de serviços sem nenhuma medida de segurança, pode facilitar o cometimento de atos ilícitos na rede, como tráfico de drogas, por exemplo. Assim, um importante instrumento de inclusão digital pode se tornar uma ferramenta para a prática de crimes cibernéticos se não dispor de mecanismos segurança que dificultem esse tipo de ação.

O objetivo deste trabalho é apresentar uma proposta de protocolo para tornar as redes sem fio públicas e gratuitas mais seguras, através da identificação de seus usuários. Dessa forma, é possível rastrear usuários que podem utilizá-las para cometer crimes cibernéticos. A proposta visa permitir uma conexão segura a esse tipo de rede, armazenando dados dos usuários para estabelecer a relação entre eles e os aparelhos utilizados para acessar a internet. Ou seja, todos os usuários que utilizam a rede sem fio, são relacionados com os seus respectivos aparelhos. Sendo assim, seria possível fornecer às

autoridades cabíveis, os dados de um usuário que utilizou um aparelho específico para realizar práticas criminosas na rede.

Esse trabalho foi dividido da seguinte maneira: na Seção 2 serão descritos os componentes criptográficos utilizados no protocolo. Na Seção 3 será apresentada a proposta de protocolo e na Seção 4 será esboçada a análise do protocolo. Na Seção 5 serão apresentados trabalhos relacionados a este projeto. Por fim, a Seção 6 apresenta os trabalhos futuros e as conclusões.

2. Componentes Criptográficos

A fim de garantir a integridade e confidencialidade dos usuários, foram estabelecidas as tecnologias para a criação da solução proposta. Essas tecnologias são apresentadas a seguir.

2.1. Criptografia Simétrica

Um algoritmo de criptografia simétrica é um tipo de algoritmo que utiliza apenas uma chave. Essa chave é utilizada tanto para encriptar um texto claro em um texto cifrado, quanto para decriptar um texto cifrado resultando em um texto claro. [Stallings 2015] explica que, um esquema de encriptação simétrica possui cinco componentes:

- **Texto claro:** É a mensagem ou dados originais, inteligíveis, que servem como entrada do algoritmo de encriptação.
- **Algoritmo de Encriptação:** realiza diversas substituições e transformações no texto claro.
- **Chave secreta:** Ela é também uma entrada para o algoritmo de encriptação. A chave é um valor independente do texto claro e do algoritmo. O algoritmo produzirá uma saída diferente, dependendo da chave usada no momento. As substituições e transformações exatas realizadas pelo algoritmo dependem da chave.
- **Texto cifrado:** É a mensagem embaralhada, produzida como saída do algoritmo de encriptação. Ela depende do texto claro e da chave secreta. Para determinada mensagem, duas chaves diferentes produzirão dois textos cifrados distintos. O texto cifrado é um conjunto de dados aparentemente aleatório e, nesse formato, ininteligível.
- **Algoritmo de Decriptação:** É o algoritmo de encriptação executado de modo inverso. Ele recebe o texto cifrado e a chave secreta e produz o texto claro original.

A notação abaixo, resume o funcionamento de um algoritmo de criptografia simétrica. O algoritmo de Encriptação é representado pela letra E. Ele é utilizado juntamente com uma chave k para cifrar uma mensagem m, resultando em um texto cifrado C. Já o algoritmo de Decriptação é representado pela letra D e é utilizado com a mesma chave k para transformar o texto cifrado C de volta em texto claro m.

$$C = E(m, k)$$

$$m = D(C, k)$$

Algoritmos de criptografia simétrica são, em geral, utilizados para garantir o sigilo de uma mensagem. Essa garantia, no entanto, depende da segurança do algoritmo empregado.

2.2. Função *Hash* Criptográfica

Uma função de *hash* criptográfica, ou apenas função *hash*, é um algoritmo que, a partir de uma entrada de tamanho variável, produz uma saída única de tamanho fixo, para aquela entrada.

Uma função de *hash* recebe uma mensagem de tamanho variável M como entrada e produz um valor de *hash* de tamanho fixo $h = H(M)$. Uma "boa" função de *hash* tem a propriedade de que os resultados da aplicação da função a um grande conjunto de entradas produzirá saídas que são distribuídas por igual e aparentemente de modo aleatório. Em termos gerais, o objeto principal de uma função de *hash* é a integridade de dados. Uma mudança em qualquer bit ou bits em M resulta, com alta probabilidade, em uma mudança no código de *hash* [Stallings 2015].

Portanto, como foi dito, uma função de *hash* garante mecanismos para verificação da integridade da mensagem enviada. Isso porque, em uma função de *hash* segura, uma mudança em um único *bit* da mensagem provavelmente o produzirá uma saída totalmente diferente da original.

2.3. MAC

Um código de autenticação de mensagem (MAC) objetiva a autenticação de mensagem entre duas partes. Diferente das funções *hash*, ela requer uma chave simétrica. Um código de autenticação de mensagem baseado em chave *hash* (HMAC) é um tipo de código de autenticação de mensagem que utiliza funções *hash* criptográfica.

Seja o seguinte cenário: um emissor A deseja enviar uma mensagem M para um receptor B, para que B tenha a garantia que a mensagem recebida tenha sido emitida por A e que seu conteúdo não foi alterado no meio do caminho, então A deve calcular um valor MAC que utiliza uma função C sobre M e uma chave secreta K de conhecimento único de A e B. Portanto:

$$MAC = C(K, M)$$

A *Keyed-Hashing for Message Authentication* (HMAC) é um exemplo de algoritmo de código de autenticação de mensagem [Krawczyk et al. 1997].

2.4. Protocolo TLS

O protocolo TLS [Allen and Dierks 1999] é um protocolo que garante a segurança de uma comunicação em redes de computadores. Ele é necessário para troca de informações na Web de forma segura, impedindo que terceiros possam capturar as informações transferidas no canal seguro de comunicação.

O protocolo permite que os aplicativos cliente / servidor se comuniquem de uma forma projetada para evitar espionagem, adulteração ou falsificação de mensagens [Allen and Dierks 1999].

Nesse protocolo, como mostra a Figura 1, o cliente TLS envia uma mensagem (CLIENT_HELLO) contendo um conjunto de algoritmos criptográficos suportados por ele, a versão TLS utilizada e uma cadeia de *bytes* aleatórios que serão utilizados em cálculos futuros. O servidor TLS responde com uma mensagem SERVER_HELLO, que

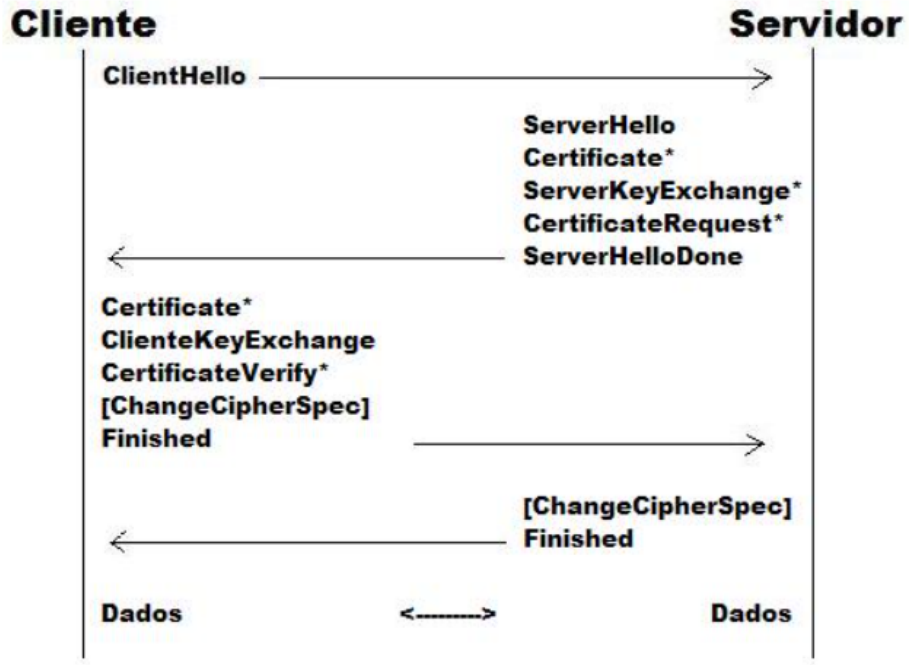

Figura 1. Processo de estabelecimento de conexão no protocolo TLS. Disponível em: `https://www.gta.ufrj.br/grad/06_1/ssl/func_tls.htm`

contém o algoritmo criptográfico escolhido na lista do cliente, o id de sessão TLS, outra cadeia de *bytes* aleatórios e um certificado digital que contém a sua chave pública. O cliente verifica o certificado digital do servidor e envia um conjunto de *bytes* aleatórios criptografados pela chave pública do servidor, que permite a ele e ao servidor calcularem a chave secreta que será utilizada na criptografia das mensagens trocadas durante a sessão. Atualmente o protocolo TLS encontra-se na versão 1.3, referenciada em [Rescorla 2018].

3. A Proposta de Protocolo

Esta seção visa descrever a proposta de protocolo. Ela apresenta de que forma os componentes criptográficos acima descritos interagem para que os usuários possam ser registrados e identificados quando necessário.

Os clientes especificados na proposta são todos os usuários que desejam se conectar a rede sem fio pública e gratuita. Para se conectar a rede sem fio, primeiramente o usuário deverá efetuar um pré-cadastro, na instituição ou no estabelecimento que fornece a rede. Após isso, ele deve efetuar um *download* da aplicação que é responsável por iniciar o protocolo. A participação no protocolo se dá por dois agentes: o cliente que deseja se conectar a rede sem fio e o servidor da aplicação.

3.1. Visão Geral

Esta subseção tem o objetivo de apresentar o funcionamento do protocolo em alto nível. Ela facilita o entendimento do mesmo, que será detalhado adiante. O funcionamento do protocolo divide-se em duas fases: o primeiro acesso e os acessos posteriores. A Tabela 1 apresenta a troca de informações, entre cliente e servidor, para liberação de acesso a um usuário utilizando a solução pela primeira vez.

A Figura 2 ilustra a iteração entre cliente e servidor, no primeiro acesso. O cliente que deseja se conectar a uma rede sem fio pública gratuita precisa primeiro estabelecer uma conexão TLS com o servidor da solução, como demonstra o passo 1.

Quando a conexão TLS é estabelecida, no passo 2 o servidor envia três códigos para o cliente, um pela própria aplicação da solução (CODE1) e dois por canais *out-of-band*, email e SMS (CODE2 e CODE3). Ao receber os códigos de verificação, o usuário envia suas informações para sua identificação, juntamente com os códigos enviados pelo servidor, passo 3. Assim, o servidor pode garantir que está recebendo as informações do cliente com o qual está conectado.

Ao receber as informações, o servidor primeiramente valida os códigos e utiliza os *web-services* disponíveis para garantir que as informações enviadas para identificação do usuário, como IMEI e CPF, são válidas. Com as informações todas validadas, o servidor armazena os dados do usuário e gera um *token* de acesso que é único para aquele usuário e o envia, passos 4.

Esse *token* tem um tempo de validade, portanto, ele será constantemente atualizado pelo servidor. O usuário recebe o *token* de acesso, e o aparelho cliente o envia para o *server* juntamente com o IMEI, passo 5. Dessa forma, o servidor tem a certeza que quem enviou o *token* é o aparelho ao qual esse *token* pertence. No passo 6, o acesso é liberado, caso todas as informações sejam válidas.

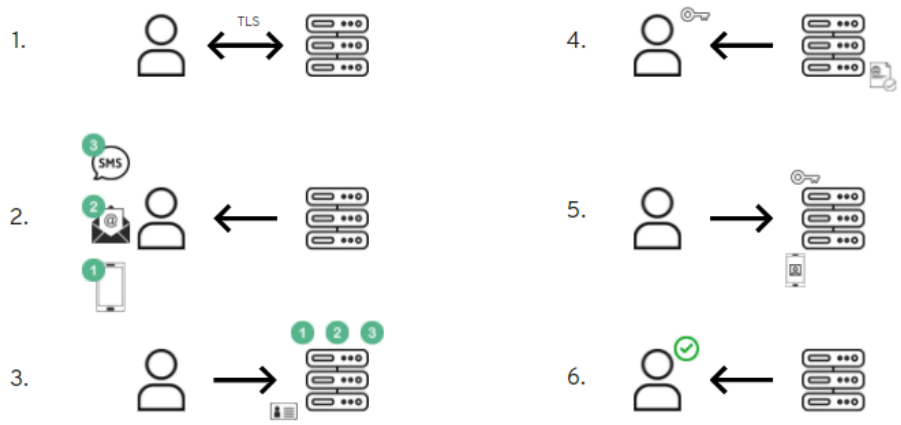

Figura 2. Comunicação entre cliente e servidor no primeiro acesso.

A cada tentativa de acesso à rede sem fio, o cliente envia o *token* e o IMEI do aparelho, e por conseguinte o servidor valida as informações enviadas, bem como o tempo de validade do *token*. Caso o servidor verifique que as informações enviadas são válidas, porém o *token* expirou, ele gera um novo *token* que é enviado para o cliente.

3.2. Modelo de Segurança

Para que o protocolo funcione corretamente e garanta a segurança e validação dos dados fornecidos pelos clientes, é assumido o seguinte:

1. Os dados do usuário, como: nome, idade e CPF, foram previamente validados, por exemplo, de forma física através da visita do usuário a um lugar apropriado. Esses dados são livres de fraudes e estão disponíveis ao sistema no momento da validação.

2. Um atacante não terá acesso a todos os canais utilizados para o envio dos códigos no primeiro acesso.
3. O número de IMEI é inalterável.

3.3. Descrição do Protocolo

3.3.1. Primeiro Acesso

Como ilustrado pela Figura 2, no passo 1 do protocolo é estabelecida uma conexão TLS, via HTTPS, entre cliente e servidor, posteriormente. No passo 2, o servidor envia ao cliente três códigos: dois usando canais *out-of-band* (email e SMS) e um utilizando a própria aplicação.

$$CODE1[SMS] \wedge CODE2[EMAIL] \wedge CODE3$$

No passo 3, o cliente retorna ao servidor uma mensagem criptografada com a chave K (que é a *TLS session key*), contendo: CODE1, CODE2, CODE3, o número de IMEI do seu aparelho (o qual é capturado automaticamente pela aplicação), número de celular, CPF, nome completo e data de nascimento. Esta mensagem é autenticada utilizando o algoritmo MAC, a chave K e CODE3:

$$[E_K(CODE3 \parallel CODE1 \parallel CODE2 \parallel IMEI \parallel Num.celular \parallel$$

$$CPF \parallel Nome \parallel Dt.Nasc)]MAC_{H(K \parallel CODE3)}$$

Ao receber a mensagem do cliente com os dados, no passo 4, o servidor primeiramente validará as informações de IMEI e CPF, utilizando os sites da Anatel e Receita Federal, por exemplo. Depois gerará um TTA (Token de Acesso Temporário) que será o *hash* da *string* concatenada do IMEI, número de celular e CPF do usuário e um número aleatório gerado pelo servidor, o qual é único.

$$TTA = H(IMEI \parallel Num.Celular \parallel CPF \parallel Num.Aleatório)$$

Ao gerar o TTA, o servidor o envia para o cliente, sempre encriptando os dados utilizando a chave K e autenticando a mensagem com MAC utilizando K e CODE3.

$$[E_K(TTA)]MAC_{H(K \parallel CODE3)}$$

Esse TTA, tem um tempo de validade e o servidor armazenará em seu banco de dados esse tempo de validade. O TTA também ficará relacionado no banco de dados, com o IMEI do aparelho para o qual ele foi gerado, para garantir que apenas esse aparelho possa utilizá-lo. Para que o usuário não precise digitar o TTA diretamente na aplicação sempre que tentar conectar na rede, ele ficará armazenado no aparelho do cliente por meio de um banco de dados também.

Tabela 1. Primeiro Acesso

1.	Cliente $\leftrightarrow$ TLS $\leftrightarrow$ Servidor	Conexão estabelecida
2.	Servidor $\rightarrow$ Cliente	CODE1[SMS] $\wedge$ CODE2[EMAIL] $\wedge$ CODE3 CODE3 = Código Aleatório gerado pelo server
3.	Cliente $\rightarrow$ Servidor	K = TLS Session Key [E_K(CODE3 $\|$ CODE1 $\|$ CODE2 $\|$ IMEI $\|$ Num. celular $\|$ CPF $\|$ Nome $\|$ Dt. Nasc] MAC$_{H(K\|CODE3)}$
4.	Servidor $\rightarrow$ Cliente	TTA = TOKEN DE ACESSO TEMPORÁRIO Cria o TTA com um *timestamp* limite TTA = H(IMEI $\|$ Num. Celular $\|$ CPF $\|$ Num. Aleatório), [E_K(TTA)]MAC$_{H(K\|CODE3)}$ Num. Aleatório é único e gerado pelo servidor.

3.3.2. Próximos Acessos Com TTA Dentro do *Timestamp*

Enquanto o TTA estiver no seu período de validade, a aplicação do usuário e o servidor seguirão os passos demonstrados na Tabela 2. A conexão TLS é sempre estabelecida no primeiro passo, afim de garantir uma conexão segura entre o cliente e o servidor.

No segundo passo, o cliente envia ao servidor uma mensagem criptografada com a chave K, contendo: o TTA, IMEI do seu aparelho (o qual é capturado automaticamente pela aplicação) e seu número de celular, assim como uma autenticação de mensagem do tipo MAC contendo o *hash* da chave K.

O servidor validará os dados recebidos no passo 3, verificando se o TTA enviado está válido ainda e se ele pertence mesmo ao IMEI daquele aparelho, e então liberará o acesso ao usuário.

Tabela 2. Próximos Acessos Com TTA Válido

1.	Cliente $\leftrightarrow$ TLS $\leftrightarrow$ Servidor	Conexão estabelecida
2.	Cliente $\rightarrow$ Servidor	K = TLS Session Key [E_K(TTA, IMEI, Num. Celular)]MAC$_{H(K)}$
3.	Servidor valida Token	Verificar MAC$_{H(K)}$ Decriptar [E_K(TTA, IMEI, Num. Celular)] Verificar relação entre TTA e IMEI no banco de dados. Verificar o *timestamp* limite do TTA.
4.	Servidor $\rightarrow$ Cliente	Libera o acesso.

3.3.3. Próximos Acessos Com TTA Fora do *Timestamp*

O cliente não saberá quando o seu TTA perderá a validade, portanto tentará se conectar ao servidor da mesma forma como demonstrado anteriormente. Porém a diferença, como pode ser observada na Tabela 3, começa na validação do servidor, no passo 3. O servidor

receberá os mesmos dados como se o TTA estivesse válido ainda, porém, ao fazer a validação de *timestamp* do TTA, verificará que o mesmo perdeu a validade e iniciará o processo de geração de um novo *token* para o cliente.

Ao verificar que o TTA não está mais válido, por *timestamp*, no passo 3, o servidor irá gerar um TTA2 para o usuário no passo 4, utilizando o mesmo método visto anteriormente. Enviará o TTA2 para o cliente que confirmará o recebimento enviando-o juntamento com o seu IMEI para o servidor, e então o servidor validará o TTA2 e liberará o acesso ao cliente.

Tabela 3. Próximos Acessos Com TTA Inválido

1.	Cliente $\leftrightarrow$ TLS $\leftrightarrow$ Servidor	Conexão estabelecida
2.	Cliente $\rightarrow$ Servidor	K = TLS Session Key [E_K(TTA, IMEI, Num. Celular)]$MAC_{H(K)}$
3.	Servidor valida Token	Verificar $MAC_{H(K)}$ Decriptar [E_K(TTA, IMEI, Num. celular)] Verificar relação entre TTA e IMEI no banco de dados. Verificar o *timestamp* limite do TTA.
4.	Servidor $\rightarrow$ Cliente	Cria um TTA2 TTA2 = H(IMEI $\|\|$ Num. celular $\|\|$ CPF $\|\|$ Num. Aleatório) [E_K(TTA2)]$MAC_{H(K)}$
5.	Cliente $\rightarrow$ Servidor	[E_K(TTA2, IMEI)]$MAC_{H(K)}$
6.	Servidor valida Token	Verificar $MAC_{H(K)}$ Decriptar [E_K(TTA2, IMEI)] Verificar relação entre TTA2 e IMEI no banco de dados. Verificar o *timestamp* limite do TTA2.
7.	Servidor $\rightarrow$ Cliente	Libera o acesso.

4. Análise do Protocolo (Esboço)

Na seção anterior foram utilizados três códigos aleatórios para cadastro de um usuário. Dois desses códigos são enviados por canais *out-of-band*, utilizando o serviços de email e SMS, respectivamente. Este método é utilizando porque, como é pressuposto pelo modelo de segurança (Seção 3.2), um atacante não terá acesso a todos os canais utilizados.

Foram concebidos possíveis cenários de ataques que o protocolo pode sofrer e de que forma ele garantiria que os dados não seriam alterados ou capturados. Também foram definidos pressupostos, no modelo de segurança, que deveram ser seguidos para o correto funcionamento do mesmo.

O primeiro tipo de ataque possível fora o de que um atacante captura o *token* de um outro celular e tenta utilizá-lo no seu aparelho para acessar a rede. Esse ataque não será bem-sucedido, pois em todas as tentativas de conexão, o aplicativo envia ao servidor o IMEI do aparelho que está tentando se conectar. Logo o servidor pode verificar que o IMEI cadastrado no banco e o IMEI do celular que está tentando se conectar são diferentes e negar a conexão.

Uma segunda tentativa de tentar burlar o protocolo seria de um usuário utilizar um CPF para se cadastrar na plataforma. Mais uma vez o ataque não terá sucesso, pois todas as informações utilizadas para identificação do usuário e do aparelho são validadas utilizando as plataformas confiáveis, como a ANATEL e Receita Federal.

Caso um atacante tentasse utilizar um ataque de replicação de mensagem, ataque no qual é capturada a mensagem do cliente contendo as informações para conexão e replicando-a para o servidor, para ganhar o acesso no lugar do cliente, ele também não teria êxito. Isso se dá pelo fato de a conexão TLS utilizar criptografia nas mensagens trocadas entre cliente e servidor. Logo, um atacante não conseguiria descriptografar a mensagem para alterar o seu conteúdo, pois ele não teria acesso a chave de sessão TLS utilizada nas trocas de menagens.

Por último, um usuário poderia tentar se cadastrar utilizando informações inválidas, exceto o número de IMEI do seu aparelho, pois este é capturado automaticamente pela solução proposta. Considerando que o requisito de segurança garante que não houve fraudulência no pré-cadastro dos usuários, esse cenário não seria possível.

5. Trabalhos Relacionados

Os trabalhos relacionados a este são, em sua maioria, sobre o gestão de identidades digitais. Tais artigos tem como foco a criação de uma solução para garantir a confiabilidade entre partes de uma transação na internet. Logo, estes artigos se relacionam com o aqui proposto no que tange a gerência de identidades dos usuários em rede, em outras palavras, garantir que o usuário é quem ele realmente afirma ser, algo necessário para esse protocolo.

Para o estabelecimento de relações comerciais, é necessário que as partes envolvidas tenham algum conhecimento umas das outras. Atender a outra parte é importante para saber qual nível de confiança nela se pode depositar e se ela é adequada, assumindo assim, se cumprirá ou não sua parte na transação [Torres et al. 2013]. Dessa forma, considerando que diversos tipos de transações ocorrem hoje em dia na internet, com tendência a aumentar ainda mais, é necessária uma solução que garanta a confiança entre as partes, sendo ela, portanto as de gestão de identidades digitais.

Contudo com o avanço tecnológico e a digitalização cada vez maior de documentos pessoais, a rede do futuro e as soluções de gestão de identidades digitais se deparam com alguns problemas para seu avanço. No novo paradigma da Rede do Futuro, muitos problemas de segurança estão relacionados ao conceito de identidade, como a explosão do número de identidades, roubo de identidade e personificação de identidade [Torres et al. 2013].

Algumas abordagens de gestão de identidades têm sido propostas, com o objetivo de fornecer assistência tecnológica para o controle e monitoramento de identidades. No entanto, como essas abordagens não consideram a segurança, confiabilidade e interoperabilidade, há uma demanda crescente por soluções, diretrizes, metodologias, ferramentas e padrões técnicos, que possam atingir essas características [Torres et al. 2013].

Contudo muitas dessas soluções propostas de gestão de identidades digitais trabalham com artefatos digitais também, como senhas, PINs, OTPs, ou certificados digitais, por exemplo, A solução proposta aqui visa utilizar a documentação real dos usuários para

identificá-los em rede, e não documentações digitais.

Para solucionar esse problema, a solução pode se estar em um aplicativo desenvolvido pelo próprio Governo Federal, o [MEUGOV 2020]. Esse aplicativo utiliza a validação de documentação real do usuário em vários níveis para liberar acesso a serviços do governo.

Esse app foi desenvolvido para facilitar o acesso a serviços digitais como SIGEPE, INSS, Receita Federal e outros por meio de um login único vinculado a uma base de autenticação alimentada com dados cadastrais pré-existentes em outras bases ligadas ao Governo Federal, como o cadastro da CNH ou até mesmo seu cadastro bancário em contas dos bancos estatais [de Albuquerque 2021].

Para acesso a diferentes serviços, exigem diferentes níveis de autenticação. O serviço do Meu gov.br separa os níveis de autenticação por selos, sendo eles: Bronze, Prata e Ouro. Como se pode imaginar, cada selo aumenta o nível de confiança na informações do usuário, dando portanto, acesso a mais serviços, sendo o selo Ouro o de maior confiabilidade.

A classe Bronze necessita somente da validação do CPF por meio da base de dados da Receita Federal. Já a classe Ouro exige a identificação inequívoca da pessoa, com acesso a dados biométricos e afins. Proporcionalmente, as permissões das diferentes classes são diferentes: enquanto a classe bronze permite apenas a visualização de dados e cadastros simples, como o Meu SUS, a classe Ouro permite acesso a modificações de dados da base da Receita Federal [de Albuquerque 2021].

6. Conclusão e Trabalhos Futuros

Como foi apresentado, as redes públicas gratuitas, em sua vasta maioria, apresentam pouca ou nenhuma segurança para conexão dos usuários. Elas não apresentam mecanismos de segurança que facilitem o rastreio de usuários mal intencionados.

A proposta de protocolo descrita visa aumentar a segurança dessas redes, de forma a auxiliar o trabalho das autoridades competentes e dos profissionais que trabalham com a manutenção e gerenciamento dessas redes. Coletando, de forma segura, os dados dos usuários conectados para poder dificultar o uso desta tecnologia para fins ilícitos.

Outra vantagem desse protocolo é de não haver a necessidade de senhas. Os usuários não precisam decorar senhas para a utilização da rede, devido a utilização de identificadores únicos para o aparelho (IMEI) e para os próprios usuário (CPF).

Apesar das contribuições trazidas pela proposta, ela apresenta certas desvantagens, como a necessidade dos usuário efetuarem um pré-cadastro fisicamente. Isso se deve para a validação mais segura dos dados, pois uma validação online poderia ser fraudada mais facilmente. Outra desvantagem é a suposição de que o número de IMEI é inalterável.

Como trabalho futuro, é necessário pesquisar soluções que tornem possível a validação automática dos dados dos usuários. Além disso, pretende-se desenvolver uma solução *mobile* que utilize o protocolo proposto para a conexão a redes sem fio públicas e gratuitas.

Referências

Allen, C. and Dierks, T. (1999). The TLS Protocol Version 1.0. RFC 2246.

de Albuquerque, R. P. (2021). Autenticação de serviços digitais via aplicativo meu gov.br. `https://csirt.ufpa.br/not%C3%ADcias/meu-gov-br`. Acesssado em: 18/05/2021.

Krawczyk, D. H., Bellare, M., and Canetti, R. (1997). HMAC: Keyed-Hashing for Message Authentication. RFC 2104.

MEUGOV (2020). Meu gov.br. `https://www.gov.br/pt-br/apps/meu-gov.br`. Acesssado em: 21/06/2021.

NAVEGAPARA (2014). Navegapará. `http://www.navegapara.pa.gov.br/`. Acesssado em: 01/06/2021.

Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446.

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

Torres, J., Nogueira, M., and Pujolle, G. (2013). A survey on identity management for the future network. *IEEE Communications Surveys Tutorials*, 15(2):787–802.