

SAAS: uma Solução de Autenticação para Aplicativos de Smartphones

Rafael Fernandes, Diego Kreutz, Giulliano Paz, Rodrigo Mansilha¹, Roger Immich²

¹Laboratório de Estudos Avançados (LEA)
Mestrado Profissional em Engenharia de Software (MPES)
Universidade Federal do Pampa (UNIPAMPA)

² Universidade Federal do Rio Grande do Norte (UFRN)
{faelsfernandes, giulliano94}@gmail.com, kreutz@acm.org
rodrigomansilha@unipampa.edu.br, roger@imd.ufrn.br

Resumo. *A crescente utilização de aplicativos em dispositivos móveis como forma de autenticação de usuários está trazendo à tona diferentes oportunidades e desafios de segurança. Por exemplo, o cartão virtual do SESC-RS utiliza um QR Code estático, contendo apenas o CPF do associado, como forma de autenticação, o que representa uma vulnerabilidade crítica sob a ótica da segurança. Neste trabalho é proposta uma solução de autenticação para aplicativos de dispositivos móveis, denominada SAAS, composta de dois protocolos principais, um de vinculação de credenciais do usuário (i.e. identificação) ao dispositivo móvel e outro para a geração de códigos de autenticação descartáveis, denominados OTACs. Como forma de demonstrar o funcionamento e a viabilidade da solução, foi implementado um protótipo que simula o controle de acesso utilizando catracas eletrônicas.*

1. Introdução

Os dispositivos móveis já fazem parte do cotidiano da maioria das pessoas. Segundo estatísticas, o Brasil possui mais de 420 milhões de dispositivos móveis conectados à Internet. Desse total, cerca de 235 milhões são *smartphones*, o que representa mais de um destes aparelhos habilitado por habitante [FGV 2019].

Uma das utilizações dos *smartphones*, ainda que incipientes, é para a autenticação (i.e. identificação e verificação) para controle de acesso (permissão ou negação de entrada) a um espaço (físico ou lógico/virtual). *Smartphones* estão sendo utilizados para identificar e permitir o acesso de usuários às instalações do SESC-RS¹ através do cartão virtual de usuário [SESC/RS 2018]. Entretanto, o cartão virtual contém um código de autenticação que é apenas um QR Code estático contendo o número do Cadastro de Pessoas Físicas (CPF). Mecanismos de autenticação como esse são conhecidos como de autenticação de único fator. O único fator é relativamente simples de ser burlado, pois basta o agente malicioso ter acesso às credenciais do usuário. Por exemplo, no caso do SESC-RS, que utiliza um único fator estático, o agente malicioso precisa apenas conhecer o CPF da pessoa ou clonar o QR Code².

Buscando mitigar alguns problemas de segurança, foram propostos protocolos de autenticação de múltiplos fatores [Di Pietro et al. 2005, Maliki and Seigneur 2007, Starnberger et al. 2009, Lee et al. 2010, Kaur et al. 2016, Ferrag et al. 2018, Wu et al. 2019].

¹<https://www.sesc-rs.com.br>

²Sabe-se que um QR Code pode ser facilmente lido a distância devido a sua própria natureza e propósito.

Uma autenticação de dois fatores pode ser realizada através da utilização de usuário/senha como o primeiro fator e um código de autenticação, gerado por uma aplicação específica (e.g., Google Authenticator) ou enviado via serviço de mensagens curtas (*Short Message Service*, SMS), como segundo fator. Entretanto, a autenticação de múltiplos fatores leva a diferentes questões e desafios de usabilidade [Cristofaro et al. 2013].

Os usuários do cartão virtual do SESC-RS querem apenas rapidamente passar pela catraca para acessar as dependências da instituição, como as academias. Nesse caso, a autenticação de múltiplos fatores é algo indesejável e, conseqüentemente, não disponível no aplicativo da instituição. A principal questão que surge é: *Como resolver o problema da autenticação, utilizando um único fator, sem comprometer a segurança e a usabilidade?*

O objetivo deste trabalho é propor uma Solução de Autenticação para Aplicativos de *Smartphones*, denominada SAAS, cuja principal finalidade é prover autenticação segura, de um único fator, sem comprometer a usabilidade. A solução é composta essencialmente por dois protocolos, um de vinculação de credenciais do usuário (i.e. identificação) ao dispositivo móvel e outro de geração de códigos de autenticação descartáveis, denominados *One-Time Authentication Codes* (OTACs). A ideia é que o usuário possa vincular a sua identidade a apenas um único *smartphone*, reduzindo a possibilidade de utilização das mesmas credenciais por múltiplas pessoas, i.e., em múltiplos dispositivos. O protocolo de vinculação gera uma chave mestra, que é utilizada pelo segundo protocolo para derivar códigos de autenticação únicos.

Este trabalho apresenta como principais contribuições: (1) projeto e discussão de um protocolo de vinculação de usuários a dispositivos; (2) o conceito e desenvolvimento de um protocolo de geração de códigos de autenticação únicos; (3) discussão de um caso de aplicação prática da solução; e (4) implementação de um protótipo da SAAS que simula a utilização da solução em catracas eletrônicas através de QR Code.

2. Os Protocolos da SAAS

A SAAS é composta por dois protocolos: identificação e verificação. O primeiro é o protocolo de identificação, onde o aplicativo do usuário é vinculado a um único dispositivo móvel. O segundo protocolo é o de geração de códigos de verificação.

2.1. Protocolo de identificação da aplicação

Após instalar o aplicativo (e.g., cartão virtual do SESC-RS), o processo inicia com alguma forma de identificação e autenticação do usuário, como login e senha. Na sequência, no primeiro acesso, é iniciado o protocolo de registro e vinculação do aplicativo. Na solução SAAS, um único dispositivo pode ser vinculado ao fator de conhecimento (e.g., login/senha) do usuário. O protocolo de identificação faz com que, em caso de uma eventual clonagem de dispositivo, a ação seja detectada de forma automática. O algoritmo detalhando este procedimento é apresentado na Tabela 1.

O processo de registro inicia com uma conexão utilizando o *Transport Layer Security* (TLS) entre o aplicativo e o servidor (linha 1). Na sequência (linhas 2 a 4), o servidor envia três códigos distintos, utilizando o canal TLS e dois canais *out-of-band* (SMS e e-mail), para o cliente. Os canais *out-of-band* são utilizados pois assume-se que o atacante

não possui os recursos necessários para comprometer todos os canais de comunicação simultaneamente.

Tabela 1. Registro da aplicação e geração da chave mestra

1. Cliente $\xleftrightarrow{\text{TLS}}$ Servidor	Conexão com verificação do certificado do Servidor
2. Servidor \rightarrow Cliente	[CODE_TLS, $code_1$]
3. Servidor \rightarrow Cliente	[CODE_SMS, $code_2$]
4. Servidor \rightarrow Cliente	[CODE_EMAIL, $code_3$]
5. Cliente, Servidor	$K_{T1} \leftarrow H(\text{tls_session_key} code_1 code_2 code_3)$
6. Cliente \rightarrow Servidor	[SEND, $nonce$, $E_{K_{T1}}(\text{imei}, \text{app_rand1})$], $\text{HMAC}_{K_{T1}}$
7. Cliente, Servidor	$K_{T2} \leftarrow H(\text{imei} \text{app_rand1} K_{T1})$
8. Servidor \rightarrow Cliente	[SEND, $nonce$, $E_{K_{T2}}(\text{server_rand})$], $\text{HMAC}_{K_{T2}}$
9. Cliente, Servidor	$K_m \leftarrow H(K_{T1} K_{T2} \text{imei} \text{app_rand1} \text{server_rand})$
10. Cliente \rightarrow Servidor	[V_MKEY, $nonce$, $E_{K_m}(\text{app_rand2})$], HMAC_{K_m}
11. Servidor \rightarrow Cliente	[V_MKEY, $nonce$, $E_{K_m}(\text{app_rand2} + 1)$], HMAC_{K_m}

A primeira chave temporária K_{t1} é gerada a partir da chave de sessão TLS e dos três códigos enviados pelo servidor. É utilizada uma função de *hash* criptográfica para gerar a chave (linha 5). A chave K_{t1} é então utilizada para cifrar (representado por E) o IMEI e o número pseudo-aleatório app_rand1 . A mesma chave é utilizada para assinar a mensagem que é enviada do cliente para o servidor (representada por HMAC). Os valores do IMEI, do app_rand1 e da chave K_{t1} são utilizados para gerar a segunda chave temporária K_{t2} . Assume-se que esta segunda chave é mais forte uma vez que esta inclui um número global único (o IMEI do dispositivo) e um número pseudo-aleatório do aplicativo (o app_rand1). Isto, naturalmente, aumenta a entropia da chave K_{t2} .

Finalmente, o servidor envia um número pseudo-aleatório server_rand para o cliente (linha 8) e ambos geram a chave mestra K_m de alta entropia (linha 9). Entretanto, ainda resta verificar a chave K_m para finalizar o algoritmo. O usuário envia um número pseudo-aleatório app_rand2 , cifrado, para o servidor (linha 10). O servidor decifra os dados recebidos utilizando a sua chave K_m , incrementa em um (+1), cifra o novo valor e envia para o cliente (linha 11). Se o cliente conseguir validar o valor recebido, significa que as chaves são iguais e a execução do protocolo é finalizada com sucesso.

Enquanto o IMEI e a chave mestra estiverem válidos ou em uso pelo usuário, não é possível registrar outro dispositivo utilizando as mesmas credenciais. O usuário só pode utilizar um dispositivo. Para mudar de dispositivo, o usuário deve primeiro revogar o aplicativo/registro atual. Este é um procedimento de segurança adotado por diferentes bancos internacionais e nacionais, como o Revolut³, N26⁴, Banco Inter⁵ e NuBank⁶.

³<https://www.revolut.com>

⁴<https://n26.com>

⁵<https://www.bancointer.com.br>

⁶<https://nubank.com.br/>

2.2. Protocolo de verificação

Um protocolo diferente para a geração de códigos únicos, que pode ser utilizado tanto para autenticação e/ou autorização, também faz parte da solução proposta. Devido ao protocolo anterior realizar a geração de uma chave mestra K_m de alta entropia do protocolo de registro do aplicativo (Tabela 1), a chave do gerador de códigos únicos pode ser derivada da chave mestra. Para a derivação, podem ser utilizadas funções de *hash* criptográfica seguras, como as dos grupos SHA2 e SHA3. A chave inicial de geração dos códigos únicos de autenticação pode ser tão simples quanto $K_c = H(K_m || K_c)$. Como a chave K_c inicia vazia, a primeira K_c é igual a $H(K_m)$.

Os OTACs, são derivados da chave K_c . Para sincronizar a geração destes códigos, é necessário a utilização dos índices iA , no aplicativo, e o iS , no servidor. No início da geração, $OTAC = K_c$ e K_c é evoluída para o próximo valor, isto é, $K_c = H(K_m || K_c)$. Já os códigos OTAC são gerados da seguinte forma: $OTAC = H^N(OTAC)$, no qual a função de *hash* criptográfica pode ser aplicada N vezes para gerar N códigos de autenticação únicos, distintos e com a propriedade de segurança denominada *Perfect Forward Secrecy* (PFS). Isto significa que não é possível descobrir os códigos OTAC passados a partir do código OTAC atual. Esta propriedade pode ser assegurada através da propriedade de irreversibilidade das funções de *hash* criptográfica.

Para exemplificar a utilização dos códigos OTAC, supondo que os índices iA e iS estão com os valores 1 e 0, respectivamente, basta o aplicativo enviar uma mensagem com o índice atual do OTAC local para realizar a autenticação no servidor. Por exemplo, o aplicativo envia a mensagem [GET, nome_arq, *nonce*, iA], HMAC para o servidor. O HMAC (assinatura da mensagem) é gerado utilizando como chave o OTAC do aplicativo. O servidor irá atualizar seu OTAC para $OTAC = H^{iA-iS}(OTAC)$, utilizando o valor do índice recebido do aplicativo. Com o novo valor do OTAC, o servidor irá verificar a assinatura HMAC da mensagem. Se a assinatura confere, o servidor irá confirmar a autenticação do aplicativo.

3. Caso de uso e desafios

Catracas eletrônicas para controle de acesso estão cada vez mais comuns. Academias, restaurantes universitários, entre outros estabelecimentos, utilizam catracas para o controle de acesso. Entretanto, a maioria segue o modelo do SESC-RS, ou seja, dados estáticos (e.g., CPF em um QR Code estático) para o controle de acesso.

Assumindo catracas simples e capazes de ler QR Code ou código de barras, OTACs podem ser utilizados para o controle de acesso. Considerando que cada indivíduo do sistema possui uma carteirinha digital de identificação, o usuário abre o aplicativo que gera automaticamente um novo QR Code contendo o novo OTAC de autenticação, como pode ser observado nas linhas 1 e 2 do algoritmo da Tabela 2.

O usuário aproxima o QR Code do leitor (linha 3) e a catraca realiza a leitura (linha 4). A catraca atualiza então o OTAC (linha 5) e verificar o HMAC utilizando o OTAC atualizado (linha 6), liberando ou não o acesso do usuário. Como pode ser observado, o processo é simples, eficiente e pode ser realizado totalmente off-line uma vez que não há nenhuma dependência de hora ou sincronização de relógios entre os dispositivos.

Vale ressaltar que códigos únicos, com alta entropia, como os OTACs, são mais

Tabela 2. OTACs em catracas eletrônicas

1. Usuário	Abre o aplicativo de identificação
2.	QR Code = [id, iA], HMAC
3.	Aproxima o QR Code do leitor da catraca
4. Catraca	Lê o QR Code
5.	Atualiza o OTAC $\leftarrow H^{iA-iS}(\text{OTAC})$
6.	Verifica HMAC utilizando o OTAC como chave

confiáveis do que dados biométricos. Em outras palavras, dados biométricos são iguais a uma senha que você não pode trocar. Se ela vaziar, compromete toda a segurança em todos os sistemas baseados exclusivamente na autenticação utilizando os dados biométricos do usuário. Na prática, autenticação biométrica deveria ser evitada como único ou principal fator de autenticação.

4. Soluções de identificação e verificação

A Tabela 3 resume diferentes soluções de identificação e verificação que utilizam dispositivos móveis, tokens, *smartcards*, entre outros recursos, como segundo ou terceiro fator de autenticação. A maioria dessas soluções são projetadas especificamente para *Internet Banking* e sistemas bancários (e.g., ATMs), focando em segurança e não em usabilidade. Exceto a solução proposta (SAAS), todas utilizam múltiplos fatores de autenticação (alguns mais simples, outros mais complexos) e são de domínio específico.

Tabela 3. Soluções de identificação e verificação robustas

Solução	Projetada para	Autenticação	Canal(is) out-of-band
[Eldefrawy et al. 2011]	<i>Internet Banking</i>	Usuário/senha e OTP via aplicação móvel	OTP é enviado para dispositivo móvel
[Pratama and Prima 2016]	<i>Internet Banking</i>	Usuário/senha e OTP via aplicação móvel	QR Code no terminal eletrônico ou PC
[Khamis et al. 2017]	Displays públicos e ATMs	Senha (PIN, ...) e dispositivo móvel	Bluetooth
[Putra et al. 2017]	<i>Internet Banking</i>	Usuário/senha, OTP, PIN e <i>smart card</i>	NFC
SAAS	Aplicativos de identificação	Autenticação de fator único utilizando OTACs	

A solução proposta neste trabalho é de propósito geral e pode ser utilizada para identificação e verificação (e.g., autenticação, autorização) em aplicativos de smartphones como o do SESC-RS. A verificação é realizada sem a necessidade de canais *out of band*. Em outras palavras, SAAS é a única solução que assegura uma verificação mais robusta utilizando apenas um único fator de autenticação. Assumindo que o acesso ao *smartphone* é seguro o suficiente, o uso da identidade do usuário estará protegida mesmo em casos de perda ou roubo do dispositivo móvel.

5. Conclusão

SAAS é uma solução de autenticação composta por dois protocolos principais, um de vinculação das credenciais do usuário ao dispositivo e outro de geração de códigos únicos. Os protocolos propostos podem ser utilizados para autenticação e autorização segura. Estes seguem os princípios dos OTACs, que promovem a geração de códigos robustos e descartáveis, ou seja, que podem ser utilizados apenas uma única vez. Isto dificulta significativamente a clonagem e a utilização dos códigos de autenticação por usuários maliciosos. O protótipo desenvolvido foi baseado nos princípios dos OTACs para realizar controle de acesso em uma catraca eletrônica. Os resultados obtidos demonstram que o custo de utilização é baixo, chegando facilmente a mais de 10 autenticações por segundo. Outra vantagem da solução proposta, diferentemente do cartão virtual do SESC-RS, é que ela utiliza códigos descartáveis para cada acesso, provendo um maior controle e segurança. Como trabalhos futuros, está previsto o estudo da aplicação da SAAS em diferentes cenários práticos, a validação formal dos protocolos da solução utilizando ferramentas de verificação automática, a evolução da solução para funcionamento tanto *online* como *off-line*, e também realizar experimentos com a solução em um cenário real.

Referências

- Cristofaro, E. D., Du, H., Freudiger, J., and Norcie, G. (2013). Two-factor or not two-factor? A comparative usability study of two-factor authentication. *CoRR*, abs/1309.5344.
- Di Pietro, R., Me, G., and Strangio, M. A. (2005). A two-factor mobile authentication scheme for secure financial transactions. In *International Conference on Mobile Business (ICMB'05)*, pages 28–34. IEEE.
- Eldefrawy, M. H., Alghathbar, K., and Khan, M. K. (2011). OTP-Based Two-Factor Authentication Using Mobile Phones. In *8th Int. Conf. on Information Tech.: New Generations*, pages 327–331.
- Ferrag, M. A., Maglaras, L. A., Derhab, A., Vasilakos, A. V., Rallis, S., and Janicke, H. (2018). Authentication schemes for smart mobile devices: Threat models, countermeasures, and open research issues. *CoRR*, abs/1803.10281.
- FGV (2019). Pesquisa Anual do Uso de TI. <http://bit.do/fgv-sp>.
- Kaur, N., Devgan, M., and Bhushan, S. (2016). Robust login authentication using time-based OTP through secure tunnel. In *3rd Int. Conf. on Comp. for Sustainable Global Development*, pages 3222–3226. IEEE.
- Khamis, M., Hasholzner, R., Bulling, A., and Alt, F. (2017). GTmoPass: Two-factor Authentication on Public Displays Using Gaze-touch Passwords and Personal Mobile Devices. In *6th ACM International Symposium on Pervasive Displays*, pages 8:1–8:9, New York, NY, USA. ACM.
- Lee, Y. S., Kim, N. H., Lim, H., Jo, H., and Lee, H. J. (2010). Online banking authentication system using mobile-OTP with QR-code. In *5th Int. Conf. on Comp. Sciences and Convergence Information Technology*, pages 644–648.
- Maliki, T. E. and Seigneur, J. (2007). A Survey of User-centric Identity Management Technologies. In *The Int. Conf. on Emerging Security Information, Systems, and Technologies*, pages 12–17.
- Pratama, A. and Prima, E. (2016). 2FMA-NetBank: A proposed two factor and mutual authentication scheme for efficient and secure internet banking. In *8th Int. Conf. on Information Tech. and Electrical Eng.*, pages 1–4.
- Putra, D. S. K., Sadikin, M. A., and Windarta, S. (2017). S-Mbank: Secure mobile banking authentication scheme using signcryption, pair based text authentication, and contactless smart card. In *15th Int. Conf. on Quality in Research (QiR)*, pages 230–234.
- SESC/RS (2018). SESC/RS lança aplicativo para acesso ao Cartão Virtual. <http://bit.do/sesc-rs>.
- Starnberger, G., Frohofer, L., and Goeschka, K. M. (2009). Qr-tan: Secure mobile transaction authentication. In *2009 International Conference on Availability, Reliability and Security*, pages 578–583.
- Wu, L., Wang, J., Choo, K. R., and He, D. (2019). Secure key agreement and key protection for mobile device user authentication. *IEEE Transactions on Information Forensics and Security*, 14(2):319–330.