Journal of Applied Research and Technology

www.jart.icat.unam.mx

Journal of Applied Research and Technology 22 (2024) 197-205

Original

2F-Authsys: A hyperlocal two-factor authentication system using Near Sound Data Transfer

D. Patel • D. Trivedi • U. Raval • A. Dennisan*

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

Received 06 10 2023; accepted 01 22 2024
Available 04 30 2024

Abstract: Two-factor authentication (2FA) is a new buzzword in the information security world. Big companies that used passwords as an authentication mechanism now have started transitioning to 2FA. One form of two-factor authentication is known as one-time password (OTP). In today's time, there is a lack of a secure multi-factor authentication platform that is also user friendly. In all the major 2FA platforms that exist today, the user has to manually enter the OTP from their phone or email. This problem is further amplified by delays in transmission of OTP via email or SMS. There has also been an increase in the number of botnets that scan large networks and try to open vulnerable ports on cloud hosting networks. An easy-to-use port security solution is not available. This research work aims to create a secure, proof-of-concept implementation of a modern and opinionated two-factor authentication system based on Near Sound Data Transfer (NSDT) technology enabling contactless secure transmission of time-based one-time password (TOTP).

Keywords: Authentication, TOTP, 2FA, NSDT, JWT

*Corresponding author.
E-mail address: daju@vit.ac.in (A. Dennisan).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

Authentication refers to the process of verifying the genuineness of a user before providing them access to a resource. It is one of the main components of cybersecurity in the present time. Some common implementations used for authentication are passwords, digital signatures, identity cards, and biometrics. Another implementation that has become very commonly used recently is the one-time password (OTP), which refers to sending a random string to a verified and trusted device or location such as the user's phone or email and asking them to enter it while logging in. This ensures that the person who is trying to log in is the same as the person who made the account and verified that the phone or email belongs to them. However, this approach is burdened by a few problems. It is hard for a user to keep track of all the OTPs that have been received. Also, it becomes difficult for the user to check when it was the last time a requested OTP is received from a particular app. The user also needs to manually enter the OTP from their email or phone into the application they are trying to log into. Additionally, the delay in receiving OTPs through SMS or email; the systems in use today are very cumbersome.

The primary aim is to create an application that does not need the user to copy and paste the OTP into the service they are using. The developed system sends the OTP transferred via sound from the mobile phone to the service that will be authenticated. Subsequently, the service will receive the respective OTP and verify whether it is the appropriate one. The security system that is being developed is a progressive web app (PWA), so it does not need to be installed onto the device. The developed secured system maintains the security standards by encrypting the OTP before it is transmitted so that it cannot be intercepted. Also, in order to prevent the replay attack, a timestamp component is added to the OTP before the encryption process so that the respective OTP will time out after a period of time.

It is very easy for botnets to scan for the open ports on virtual private servers (VPS) and other servers, especially in large cloud infrastructure service providers because they use internet protocol (IP) addresses in a certain range. The developed system aims to prevent the unauthorised scanning by providing a port security solution that uses port knocking to open and close ports on a server via a webapp. Once a user requests access to a particular port, only that user can see that port as open by 'port knocking', and after one connection, it expires as well. This port security mechanism can reduce the effectiveness of the reconnaissance phase in cyberattacks. Therefore, a time-based one-time password mechanism using Near Sound Data Transfer that allows access to a port-knocking service for the servers is implemented as a use case example.

2. Literature survey

A zero effort two-factor authentication system (AlQahtani et al., 2020) has been proposed based upon the data that the user has. An authentication factor can be defined based upon passwords, data in smart cards, smartphones, fingerprints, irises, as well as keystrokes. Here, the data from the broadcast messages are used to implement a two-factor authentication system to determine whether two devices are proximate or not to ensure that the respective device belongs to the same user. The very idea of safeguarding valuable and sensitive information has been considered mandatory and has been around us for many centuries now. These days, modern computer systems, as well as the installations, have deployed an effective authentication system in order to safeguard the machine and the installation from unauthorised access by intruders. An average user holds more than 20 professional and personal accounts, and unfortunately these users solely depend on their usernames and passwords for authentication (Bachmann, 2014). Due to the recent trend towards data breaches, it is mandatory and significant that future security holds in multiple authentication processes such as sound-oriented passwords, electronic tattoos and, of course, biometrics. A proximity detection system based on sound (Choi et al., 2018) has been proposed to prevent the relay attacks on a passive keyless entry and start (PKES) system in recent cars. This very idea has been thought of since it is very common that a modern car has the audio system inbuilt. Considering the scenario where cars are parked commonly, the developed model has been tested for its usability and security. Also, the record-and-play attack has been demonstrated to prove that the developed model is robust to the respective attack.

A novel source coding scheme for authenticating audio signal based on chaotic dynamic system as well as set partitioning in hierarchical trees (Fan, 2020) is proposed. The chaotic dynamic system is capable of anti-synchronization counterfeiting attack and self-recovery. Based on the position and content of the audio, check bits are generated by hash algorithm and the chaotic sequence to make the system robust. The roles played by the various authentication systems on the internet are considered as how the users are being digitally divided and excluded. The primary factors that are considered and highlighted are psychological as well as material and skill-oriented barriers (Gibson et al., 2010). And it is observed that a number of disagreements of accessibility and security goals are exhibited in image- and sound-based authentication systems. These days, web authentications are based on the challenge response model (Guillet et al., 2010) that evades the obvious transmission of the passwords over the network. Even though this model increases the protocol exchanges, most of the existing protocols integrate the model

with it to authenticate a user for providing service. At the same time, it minimises the handshakes.

One of the primary and significant protocols to secure the communication on the internet is the transport layer security (TLS). An improvised version to TLS, called the TLS-HOTPS protocol (Hamdane et al., 2011), is proposed to provide client authentication, which uses the pre-shared keys through the HMAC-based one-time password method. The improvisation ensures the protection and security of the client credentials. Also, an appropriate validation of protocol's security accomplishment is provided. Several resources and services need protection and security from illegitimate use, especially with the distributed system. A generic as well as a secured system that elevates the two-factor authentication to three-factor authentication system (Huang et al., 2011) is proposed. A three-factor authentication mechanism (Idhom et al., 2020) such as passwords, biometrics, and smart cards is investigated in a systematic manner to authenticate the users. Securing the port access is one of the critical and significant points that has to be considered in providing robust security to the computer system. The very problem occurs when a user is not properly authenticated when accessing the ports. A detailed study is performed on three different ports such as port 22, port 23, and port 80 (Jia et al., 2008) with respect to its secured access. Here, port 22 refers to SSH, port 23 refers to telnet, and port 80 refers to web. Based upon the result analysis and subsequent tests, it is observed that the security of the network systems has been improvised both in terms of local and public networks. Also, it is observed that the SSH protocol has a time difference of 2.42 seconds, telnet with a time difference of 2.14 seconds, and web with a time difference of 2.19 seconds.

An improvised universally composable password-oriented key exchange protocol (Khader et al., 2016) is proposed. In order to reduce the consumption of bandwidth in communication, the one-time signature is replaced with the message authentication codes. It is noticed that the improvised method saves as much as 12 Kb of bandwidth in its implementation. A new covert channel for stealthy communication is proposed. Here, least significant bit (LSB) steganography as well as Tariq port knocking (Li, 2010) is utilised by the respective communication channel to hide the sensitive data. Also, before hiding the sensitive data, the GnuPG encryption method is deployed over it as an additional layer of protection. The communication efficacy is examined, and it is observed that the channel achieves 152 bps as a maximum rate of transmission. A new method that utilises a combination of both the signatures as well as the pronounced name utterances (Malikovich et al., 2019) is proposed to authenticate the users. Here, the dynamic features of digital signatures are extracted for authentication rather than the traditional signature verification methodology. And it is observed that the proposed method provides a relatively simple as well as useful mechanism for user identity and authentication. In order to find solution to eavesdropping as well as the replay attack, a simple and useful one-time password (OTP) mechanism is utilised. Here, a method that uses the pseudorandom number generators along with the OTP (Ali et al., 2012) is proposed to make the authentication process effective.

A new method that simplifies the port-knocking mechanism is proposed by utilising the source port sequences. A mechanism that controls the start and stop of certain services is introduced to minimise the issues that arise during the port scanning as well as the TCP replay attack. Based on the evaluation performed, it is observed that the proposed methodology performs better and faster when compared to other similar methods. A TLS standard and asynchronous OTP mechanism (Msahli et al., 2015)-based flexible authentication system is presented for safe box cloud service. Additionally, HTOP which preserves the maximum reliability to TLS protocol is considered. A means of sequence determination that contains the pre-shared key hash, client IP addresses and destination port, and time value (Popeea et al., 2011) is considered to accomplish the client-server synchronization. Based on the synchronization of client-server, the vulnerabilities can be mitigated to a larger extent. An execution of port knocking with the help of x509 certificates (Sel et al., 2016) is introduced to achieve high scalability with respect to port knocking. As it is known, port knocking hides remote services that are behind a firewall; here, only after successful authentication to the firewall access to the listening ports.

A secured knock sequence using AES encryption mechanism (Srivastava et al., 2011) is implemented to overcome the existing vulnerabilities and attacks such as replay attack, man-in-the-middle attack, out-of-order packet delivery, and use of spoofed packets. One of the significances of the proposed method is that by means of sniffing and spoofed packet utilisation, it cannot be detected. The usage of universal serial bus communication has indeed become an essential digital commodity these days for external storage and transmission. Despite its advantages, there is no in-built mechanism to authenticate the users. Here, a two-factor authentication mechanism (Tritilanunt et al., 2014) is proposed to alleviate the existing problem, which results in non-copying and distribution of software. A secured zero-effort two-factor authentication is proposed based upon two types of ambient audio signals called SoundAuth (Wang et al., 2018). By having smartphones and browsers, the proposed system looks for the signs of proximity by comparing their surrounding sounds. Here, the machine learning algorithms are utilised to analyse the audio signals. A quick authentication method called QuickAuth (Zhu et al., 2016) is proposed to be implemented with no dependence to the password and with less user involvement. One of the authentication factors in the proposed system is the user's

smartphone, and the other authentication factor is the proximity of the respective smartphone towards the computer through which the user logs in. The proximity of both devices is acquired through the microphones.

3. Overall architecture

The port-knock backend is a Golang service that listens to/sends a special knock TCP packet and opens the requested port, thereby whitelisting the IP addresses that are requested by the port knock. The database and the port-knock backend are connected by the ExpressJS backend container, which primarily contains most of the logic for the application. It provides the JWT tokens, validates TOTPs, contains the public and private keys for encryption, and manages the basic backend functionality of a website. It is capable of handling a lot of concurrent requests since it runs on process manager 2 (PM2). The PM2 is a production Node.js process manager with features such as automatic application load balancing, declarative application configuration, deployment system, and monitoring. It also supports multithreading and multiprocessing.

When the ExpressJS backend gets a validated request to open a port, it automatically connects to the port-knock backend to send a specifically crafted TCP packet to knock a port open using the knock backend. This opens the port on the server running another instance of knock backend for a particular IP address that requested the port opening request (knock).

The frontend is created using React and is a progressive web application (PWA) accessible on mobile as well in the form of a native application. A reverse proxy is used to route all the traffic to the correct places such as /mobile to serve the PWA, and /api/ will route the traffic to the ExpressJS backend. All the components are dockerized so that only the reverse proxy container is exposed to the web server, and the web server is in turn exposed to the internet. The reverse proxy also handles all the HTTPS traffic since the HTTPS certificates generated are stored on the server. This ensures security and allows microphone access on browsers like Chrome, which require HTTPS to give permission to use the microphone audio. Fig. 1. Exhibits the overview architecture of the proposed authentication system, 2F-AuthSys.

3.1. Enrolment process

The first step for the user is the enrolment process, as shown in Figure 2. The initial signup process consists of Google authorisation, which takes the user through a stepwise process to add their device for the two-factor authentication mechanism.

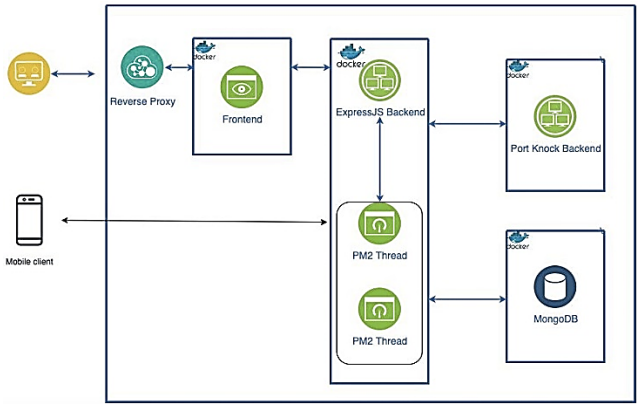

Figure 1. Architectural overview of the proposed authentication system.

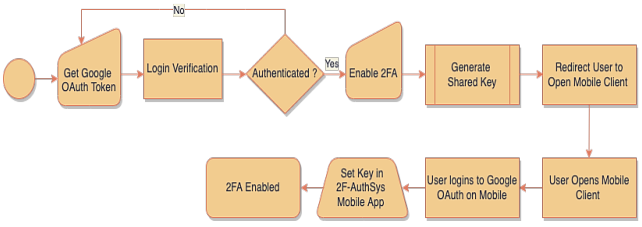

Figure 2. Enrolment process of 2FA AuthSys.

Step 1: Signup/login with a Google account. The respective user is registered to the service and subsequently, 2FA is enabled by generating a secret key.

Step 2: Scan the QR code on the phone or use the provided weblink for logging in.

Step 3: If the user is registered already, they will be directed to the verification page for logging in directly.

Step 4: For the first time user, TOTP is fetched from the server, and the respective device is added. The website listens for the TOTP transmitted from the mobile device using audible sound and validates the device.

Step 5: After successful verification, the respective access token is stored in the browser, and the user is taken.

3.2. Google OAuth process

The Google OAuth mechanism shown in Figure 3 involves interactions between client-side application and Google's authorization servers.

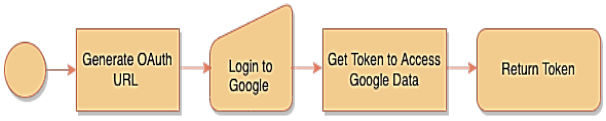

Figure 3. Google OAuth mechanism.

Using the Google API, the client application requests access from the Google authorisation server, thereby extracting an unauthorised token from the respective response. If the user is not already logged in, then Google prompts the user to log in. Google then displays an authorisation page that allows the user to see what Google service data their application is requesting to access. If the user approves their application's access request, Google issues an authorised request token and sends the token to the respective Google API that they want to access. A refresh token is used to enable the access tokens having the limited lifetimes.

3.3. FA login process

Once the registration of the user is performed, the respective user logs in to utilise the services. After the Google OAuth token is generated, a login verification from mobile and desktop is generated on both devices, which is a time-based code using UNIX time.

The code expires every 30 seconds and regenerates again. Using the library, the TOTP is sent to the transmitter that will send audible tones through mobile device speakers and a receiver (desktop client) that subsequently decodes the tones through the microphone. Using the shared key, the TOTP token is verified from the backend and the authentication token is set. The two-factor authentication login process is shown in Figure 4.

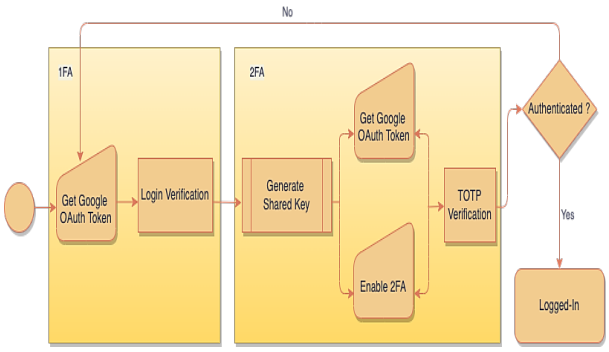

Figure 4. 2FA AuthSys login process.

3.4. TOTP algorithm

While performing the 2FA authentication, a secret key is generated that is stored for that specific user in the database. When a user is being authenticated, the secret key is requested from the server and is converted to audio once the send button is activated on the mobile device. On desktop client, the audio is decoded and converted to text which is verified with the secret key and current time using the getTOTP method from the server. The user is authenticated successfully when OTP matches correctly and under 30 seconds time constraint. Figure 5 shows the flow of the TOTP algorithm.

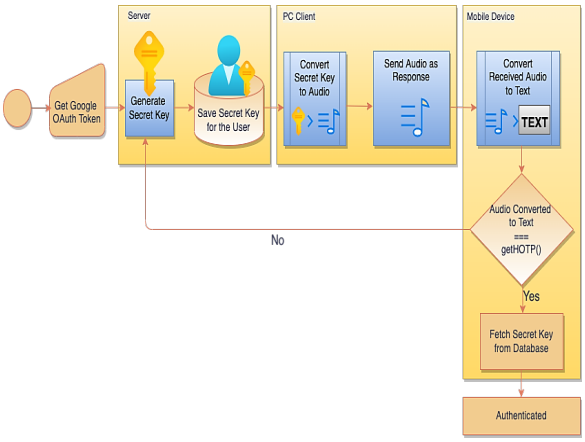

Figure 5. TOTP algorithm flow diagram.

3.4.1. JWT flow

The basic JWT flow is shown in Figure 6. Once the user is logged in, each subsequent request includes the JWT in HTTP header, allowing the user to access routes, services, and resources that are permitted with that token. The JWT is signed using a secret key (with the HMAC algorithm). The JWT token is stored in the browser local storage and validated whenever the user makes a request to the server, if the token is expired, the token from local storage is deleted, and the user is required to login again to get a new token.

3.4.2. JWT authentication

The JWT token authentication mechanism is modified to obtain a valid component that is implemented using a refresh token. The refresh token adds a second layer of security and is for a long-term access (one week expiry) to an API on behalf of the user. In such long-term scenarios, the user is not always present. Hence, the refresh token allows an application to autonomously obtain a new access token from the security token service, without user intervention. In case the refresh token is expired, the website asks for a re-login via Google OAuth, as shown in Figure 7. After successful authentication with the mobile device, the JWT access token is upgraded to an access 2FA token, which is required to verify TOTP.

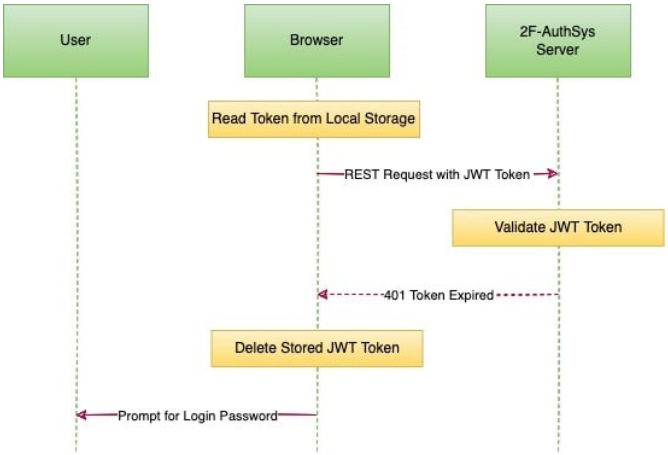

Figure 6. Basic JWT token authentication.

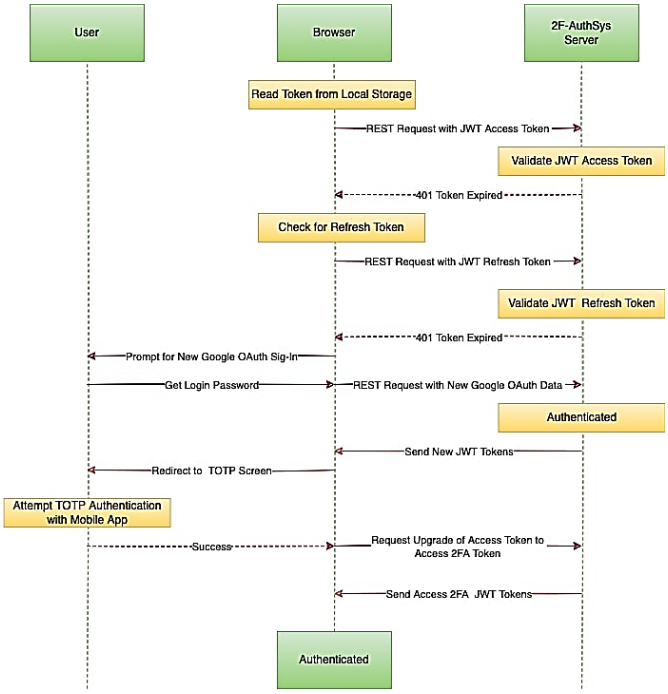

Figure 7. Modified JWT token authentication.

4. Results

The web app successfully implements the TOTP based authentication algorithm as well as the port-knocking feature, which opens the ports on the backend server. The server and web application were extensively tested for security using tools such as SQLMap, Nikto, Nmap, Netcat, and Gobuster. The SQL map tool, which is an open-source penetration testing tool that automates the process of detecting and exploiting through the SQL injection and subsequently takes over the database servers, is shown in Figure 8. It is observed that the SQL map test results show that there are no SQL injection vulnerabilities present in the application.

Figure 8. 2F-AuthSys development process.

The result of a Nmap scan to check for open ports on the backend server and verify if the port-knocking mechanism was working properly or not, is shown in Figure 9. As expected, only ports 22 (SSH), 80 (HTTP), and 443 (HTTPS) were open without knocking.

Nikto is an open-source web server scanner that performs tests for multiple items, including potentially dangerous files/programs, and checks for outdated versions of servers and version specific problems, as shown in Figure 10. The respective tool has been used to check for any vulnerabilities in the web server and subsequently could not find one. It only raised a warning about XSS headers not being defined, but this was verified and patched out as well. This can be checked in the report from securityheaders(dot)com.

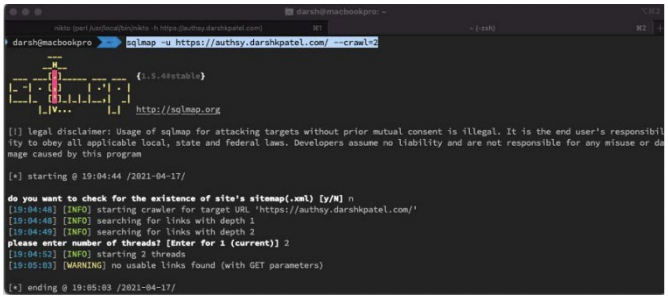

Figure 9. Vulnerability check through SQL map scanning.

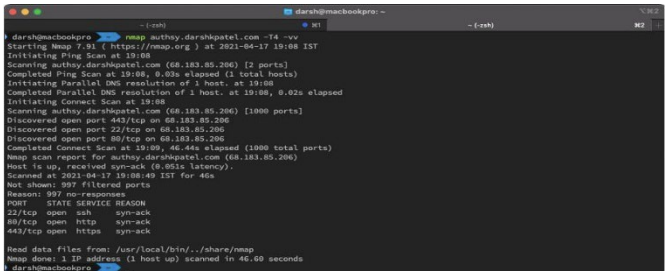

Figure 10. Verification of port-knocking mechanism through Nmap scan.

Figure 11 shows that the website where the project has been deployed has proper security headers configured.

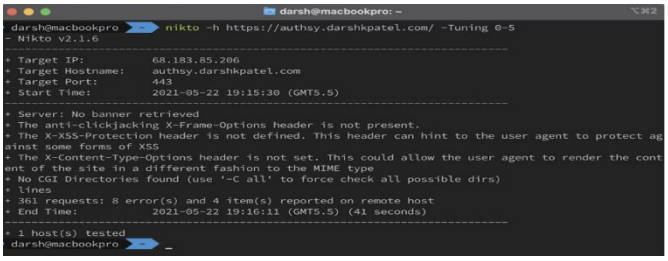

Figure 11. Nikto scanner.

Figure 12 shows the scanning report of authsy.darkspatel(dot)com from the securityheaders(dot)com webportal.

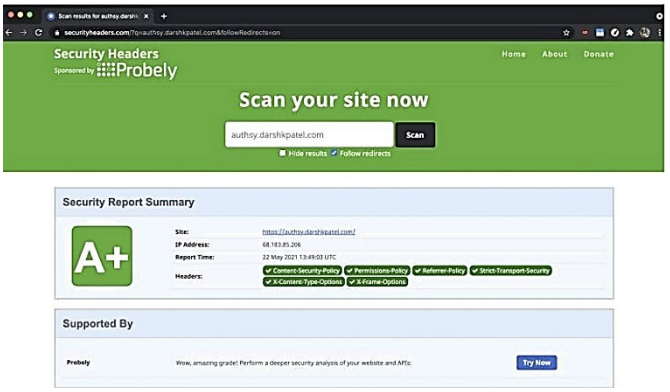

Figure 12. Securityheaders(dot)com scanning report.

5. Result analysis

The accuracy of the TOTP transmission from mobile to PC is 100% accurate and every time when the OTP is generated with the correct accounts, the system succeeds in picking it up and authorising access. A comparative analysis of various 2FA algorithms, including the proposed method, 2F-AuthSys, is given below in tabular form. The value '0' is interpreted as 'false' and '1' as 'true' for the intersecting case in the table.

A comparative analysis of various authentication algorithms along with the proposed system is tabulated in Table 1. The comparison is performed based upon the usability, deployability as well as the security of the authentication methods.

Table 1. Comparative analysis of the proposed system (2F-AuthSys) with other 2FA algorithms.

Methods	Usability								Deployability								Security							
	Memorywise-effortless	Scalable-for-User	Nothing-to-Carry	Physically-effortless	Easy-to-Learn	Easy-to-Use	Infrequent-Errors	Easy-Recovery-from-Loss	Accessible	Negligible-Cost-per-User	Server-Compatible	Browser-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Throttled-Guessing	Resilient-to-Unthrottled-Guessing	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-orVeri	Resilient-to-Phishing	Resilient-to-Theft	No-Trusted-Third-Party	Resilient-to-Explicit-Consent	Unlinkable
SV-2FA	1	1	1	1	1	1	0	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	0	1	0
Password	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1
Google-2-Step	0	0	1	0	1	1	1	1	0	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1
QuickAuth	0	0	1	0	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	0	1	0
SoundAuth	0	0	1	0	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	0	1	1	0	0	0
Sound-Proximity	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	0	1	1	1	0	1	0	0	1	1
3FA	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0
2F-AuthSys (Proposed)	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0

6. Conclusion and future work

A hyperlocal two-factor authentication system using Near Sound Data Transfer and port-knocking based web server security named as 2F-AuthSys is proposed and implemented. The developed security system is compared with other similar systems based on three parameters such as usability, deployability, and security. It is observed that the proposed security system outperformed all other existing security systems based upon the parameters that are defined.

After the security analysis of the web application and the server, the application was deployed at authsy(dot)darshkpatel(dot)com, which is a domain owned by one of the authors. The website functioned well in all the conducted reviews and was kept online until 10 January 2022. Some future enhancements could include push notification on the mobile PWA when the service is waiting for the TOTP. A history of ports knocked by a user in their dashboard would also be useful so that the user does not need to manually enter the ports each time.

The customisation of ports that can be whitelisted would be another future work that can be achievable. The backend already supports blacklisting of JWT tokens, but the frontend interface could be implemented so that a user can report stolen tokens, which will then be blacklisted.

Conflict of interest

The authors have no conflict of interest to declare.

Funding

The authors received no specific funding for this work.

References

AlQahtani, A. A. S., Alamleh, H., Gourd, J., & Mugasa, H. (2020). 0ei2fa: Zero effort indoor two factor authentication. In *2020 14th International Conference on Innovations in Information Technology (IIT)* (pp. 253-257). IEEE. https://doi.org/10.1109/IIT50501.2020.9299049

Ali, F. H. M., Yunos, R., & Alias, M. A. M. (2012). Simple port knocking method: Against TCP replay attack and port scanning. In *Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)* (pp. 247-252). IEEE. https://doi/10.1109/CyberSec.2012.6246118

Bachmann, M. (2014). Passwords are dead: alternative authentication methods. In *2014 IEEE Joint Intelligence and Security Informatics Conference.* The Hague, Netherlands, (pp. 322-322). IEEE. https://doi/10.1109/JISIC.2014.67

Choi, W., Seo, M., & Lee, D. H. (2018). Sound-proximity: 2-factor authentication against relay attack on passive keyless entry and start system. *Journal of Advanced Transportation*, *2018*, 1-13. https://doi.org/10.1155/2018/1935974

Fan, M. (2020). A source coding scheme for authenticating audio signal with capability of self-recovery and anti-synchronization counterfeiting attack. *Multimedia Tools and Applications*, *79*(1), 1037-1055. https://doi.org/10.1007/s11042-019-08095-x

Gibson, M., Conrad, M., Maple, C., & Renaud, K. (2010). Accessible and secure? Design constraints on image and sound based passwords. In *2010 International Conference on Information Society* (pp. 423-428). IEEE. https://doi/10.1109/i-Society16502.2010.6018741

Guillet, T., Moalla, R., Serhrouchni, A., & Obaid, A. (2009, December). SIP authentication based on HOTP. In *2009 7th International Conference on Information, Communications and Signal Processing (ICICS)* (pp. 1-4). IEEE. https://doi.org/10.1109/ICICS.2009.5397549

Hamdane, B., Serhrouchni, A., Montfaucon, A., & Guemara, S. (2011). Using the hmac-based one-time password algorithm for tls authentication. In *2011 Conference on Network and Information Systems Security,* La Rochelle, France. (pp. 1-8). IEEE. https://doi/10.1109/TPDS.2010.206

Huang, X., Xiang, Y., Chonka, A., Zhou, J., & Deng, R. H. (2010). A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, *22*(8), 1390-1397. https://doi/10.1109/TPDS.2010.206

Idhom, M., Wahanani, H. E., & Fauzi, A. (2020). Network Security Applications Using the Port Knocking Method. In *Journal of Physics: Conference Series* (Vol. 1569, No. 2, p. 022046). https://doi/10.1088/1742-6596/1569/2/022046

Jia, H. Y., Qing, S. H., Gu, L. Z., & Yang, Y. X. (2008). Efficient Universally Composable Password-Based Key Exchange. In *2008 International Conference on Computational Intelligence and Security* (Vol. 2, pp. 293-298). IEEE. https://doi/10.1109/CIS.2008.148

Khader, M., Hadi, A., & Hudaib, A. (2016). Covert communication using port knocking. In *2016 Cybersecurity and Cyberforensics Conference (CCC)* (pp. 22-27). IEEE. https://doi/10.1109/CCC.2016.12

Li, F. F. (2010). Sound-based multimodal person identification from signature and voice. In *2010 Fifth International Conference on Internet Monitoring and Protection* (pp. 84-88). IEEE. Barcelona, Spain. https://doi/10.1109/ICIMP.2010.18

Malikovich, K. M., Turakulovich, K. Z., & Tileubayevna, A. J. (2019). A method of efficient otp generation using pseudorandom number generators. In *2019 International Conference on Information Science and Communications Technologies (ICISCT)* (pp. 1-4). IEEE. https://doi/10.1109/ICISCT47635.2019.9011825

Msahli, M., Hammi, M. T., & Serhrouchni, A. (2015). Safe box cloud authentication using TLS extesion. In *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)* (pp. 1-6). IEEE. https://doi/10.1109/SSIC.2015.7245679

Popeea, T., Olteanu, V., Gheorghe, L., & Rughiniş, R. (2011, June). Extension of a port knocking client-server architecture with NTP synchronization. *In 2011 RoEduNet International Conference 10th Edition: Networking in Education and Research* (pp. 1-5). IEEE. https://doi.org/10.1109/RoEduNet.2011.5993704

Sel, D., Totakura, S. H., & Carle, G. (2016). sKnock: port-knocking for masses. In *2016 IEEE 35th Symposium on Reliable Distributed Systems Workshops (SRDSW)* (pp. 1-6). IEEE. https://doi/10.1109/SRDSW.2016.11

Srivastava, V., Keshri, A. K., Roy, A. D., Chaurasiya, V. K., & Gupta, R. (2011, April). Advanced port knocking authentication scheme with QRC using AES. In *2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)* (pp. 159-163). IEEE. https://doi.org/10.1109/ETNCC.2011.5958506

Tritilanunt, S., Thanyamanorot, N., & Ritdecha, N. (2014). A secure authentication protocol using HOTP on USB storage devices. In *2014 International Conference on Information Science, Electronics and Electrical Engineering* (Vol. 3, pp. 1908-1912). IEEE. https://doi/10.1109/InfoSEEE.2014.6946255

Wang, M., Zhu, W. T., Yan, S., & Wang, Q. (2018). SoundAuth: Secure zero-effort two-factor authentication based on audio signals. In *2018 IEEE Conference on Communications and Network Security (CNS)* (pp. 1-9). IEEE. https://doi/10.1109/CNS.2018.8433202

Zhu, X., Yu, S., & Pei, Q. (2016). QuickAuth: two-factor quick authentication based on ambient sound. In *2016 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). IEEE. https://doi.org/10.1109/GLOCOM.2016.7842192