Progression Towards a Safer and Private Authorization for Business Entities: Fire OAuth

D. Bala Gayathri(✉), M. Tharunraj, Rozen Berg(✉), and M. Sanjay Kannan(✉)

Department of Information Technology, Anna University, Chennai, India
gayathribalansvg@gmail.com , rozenberg4christ@gmail.com ,
sanjaykannanmurali2001@gmail.com

Abstract. There is growing interest in alternatives to conventional techniques like username and password combinations due to the growing requirement for safe and effective authentication and authorization systems. Token-based solutions, like the OAuth framework, are one such possibility. But there are a number of problems with how OAuth is currently implemented, including poor performance, a lack of privacy, and vulnerability to security risks. These problems are addressed by the revolutionary token-based authorization method called Fire OAuth that we provide in this work. Our technology offers quicker, safer, and more private authentication and authorization by combining cryptographic methods with a decentralised design.

Keywords: ExpressJS · HMAC · Magic Link · Micro service · OAuth · Private · Session Sync · Single-use tokens · WebSockets

1 Introduction

More and more people use public computers (like Web terminals) to do business online. But using a password or username to access today's web-based email, online marketplaces, or banking websites is always necessary to authenticate oneself with the distant provider. People rely increasingly on digital systems to conduct online commerce as computing becomes more commonplace. An essential enabler for many applications is security. Financial institutions should utilise efficient techniques to verify the identities of clients who use Internet-based goods and services. Today's web-based services always demand you to provide your username and password in order to log in. Since the password may be obtained via the shared computer and then utilised by an adversary, this is a serious vulnerability. The user authentication offered by the present online payment systems is insufficient.

Any computer system that deals with sensitive data must have authentication and permission. These procedures are in charge of making sure that users' identities are confirmed and that only authorised people have access to particular resources. The demand for reliable and secure identification and authorization

S. Jayasingh et al. (Eds.): ICETBM 2023, AEBMR 242, pp. 243–257, 2023.
https://doi.org/10.2991/978-94-6463-162-3_22

systems has grown significantly as a result of rising internet usage and the proliferation of linked devices.

In the past, username and password combinations have been used to accomplish authentication and authorisation. However, security risks including phishing, social engineering, and brute force attacks have made these techniques more and more susceptible. Additionally, the usage of straightforward username and password combinations can result in a bad user experience because users are frequently forced to remember numerous passwords and run the risk of being locked out of their accounts if they forget them.

We frequently require one service to communicate on our behalf with another. Let's look at two situations:

1. Alice has a contact list with hundreds of people on her Gmail account. She signs up for Facebook and wants to know which of her Gmail connections she may friend there. She can check Facebook for each one separately, but it would be lot simpler if Facebook could read her Gmail interaction between users.
2. Bob wants to utilise a picture printing business to print every photo in his private Picasa album and ship the hard copies to his grandparents. The photographs are from a family gathering. Of course, Bob could print them all independently, but it would be preferable to point the print company to his Flickr album and have the images printed there instead.

As a result, alternatives to conventional techniques of authentication and authorization are gaining popularity. Token-based systems are one of these alternatives. By using a token rather than a password to access resources, token-based systems, like OAuth, offer a more effective and safe way to authenticate users and grant access to resources. As a digital representation of the user's identity, the token, which is often a string of characters, can be used to access resources on the user's behalf. By using the open standard for permission known as OAuth, users can offer access to their resources to other applications without disclosing their username and password. It is widely utilised in a variety of web-based services, including cloud storage services and social networking platforms. But there are a number of problems with how OAuth is currently implemented, including poor performance, a lack of privacy, and vulnerability to security risks.

As the process of obtaining and confirming tokens might take some time, slow performance is a major problem with existing OAuth implementations. This may result in slower access to resources and a less satisfying user experience. Furthermore, current OAuth implementations lack privacy because tokens are long-lived and is used multiple times without the user knowing it. This may result in sensitive data being compromised and unauthorised access to resources. Open Authorisation, often known as OAuth, is a free standard for secure authentication and authorization of users that is increasingly being used in a wide range of new industries. Without disclosing their identities and passwords, users may authorise access to their information to apps from third parties via OAuth. As a result, users' comfort is increased and they no longer need to remember several login passwords for various services, which not only increases security.

The internet of things (IoT) and home automation are two important areas where OAuth is in use. OAuth enables safe and simple access to the increasing amount of devices in homes without requiring users to generate and remember several login passwords. This is crucial for consumers who wish to utilise a phone device or web portal to remotely operate their smart devices. OAuth is also being used in the developing field of digital health. Without disclosing login information, OAuth allows patients to securely exchange their health information with healthcare providers. This enables better patient outcomes as well as more effective and efficient healthcare delivery. OAuth is also utilised in the area of financial technologies (FinTech) to provide easy financial services like mobile payments and online banking. Users may now safely view their financial data and conduct transactions without disclosing their login information.

2 Literature Survey

2.1 Background

Any computer system that manages sensitive information must have both authentication and authorisation. While permission is the process of giving access to particular resources based just on user's identification, authentication is the procedure of confirming the identity of a user. Together, these procedures make sure that only those with permission may access private data and resources.

2.2 Traditional Methods of Authentication

Historically, username and password combos have been used to accomplish authentication. Although this authentication technique is popular and easy to use, it has a number of drawbacks. Its susceptibility to security risks including phishing, psychological manipulation, and brute force assaults is a serious drawback. Phishing is a form of social engineering assault where the attacker poses as a reliable party in order to deceive the victim into disclosing their login and password. Brute force attacks include a huge number of potential combinations being tried in an effort to guess the user's password [7]. The fact that traditional authentication techniques frequently result in a bad user experience is another drawback. Users frequently need to remember many sets of credentials, and if they miss their passwords, they risk having their accounts closed. Furthermore, using straightforward usernames and passwords combinations might result in inadequate protection since users may utilize the same password for several accounts, leaving them open to password-based assaults [4].

By giving the most practical and secure organisation a password-protected verification mechanism. The password served as the most effective form of authentication in the past for limiting illegal access. As innovation in the authentication process advances, passwords are now modified to be more secure. The industry and academics have been compelled to discover a new alternative since this conventional approach is vulnerable to threats including theft, hacking, and

password cracking. This research attempts to describe the specifics and workings of each technique and goes into further depth about the main methods for determining a password's validity [6,8]. Users who use services like Dropbox or Google Drive to store their personal data are increasingly turning to the cloud for storage. The encryption key used to encrypt submitted data should preferably be managed and only accessible by the user for security reasons. Current encryption technologies either demand that the user manage secure cryptographic keys or generate keys from flimsy passwords [2]. While the second strategy is more practical but provides only little protection since data is encrypted is vulnerable to offline assaults, the former approach has significant usability concerns and necessitates safekeeping by the user. With the use of a distributed server arrangement, the recently developed idea of password authenticated secret-sharing enables users to safely generate strong keys from weak passwords effective encryption However, PASS is not as ideal for encryption as first believed because it only takes into account the construction works of a single, static key, but actual encryption would call for the management of multiple keys [9,12].

2.3 Emerging Business Trends for Password Authentication

An rising trend in business and management [5] is the use of bio-metric technologies, such as fingerprint and face recognition, for password authentication. Compared to conventional text-based passwords, this authentication approach is more secure since bio-metric data is specific to each user and cannot be easily copied. Users may find bio-metric technology more convenient as a result of not having to remember several text-based passwords. Additionally, a lot of businesses are now using password management technologies that let consumers manage and securely save their credentials across many devices and accounts [1]. The usage of multi-factor authentication, which combines conventional text-based passwords with other types of verification like a fingerprint scan or a one-time code given to a mobile device, is another trend in business and management for password authentication. A better level of security is offered by this method since it is more challenging for hackers to access numerous authentication methods. Additionally, a lot of businesses now employ risk-based authentication, which considers both the context of the authentication request and the possibility of a false request. This enables businesses to see possible hazards earlier and take prompt, appropriate action [3].

Businesses are also aiming to develop solutions that enable users to authenticate themselves across numerous devices and platforms due to the growing popularity of virtualized services and mobile phones. This includes the usage of Identity Management solutions, which let users log in only once to access a variety of programs and services without repeatedly entering their passwords. The usage of password - based authentication solutions, which completely do away with the necessity for text-based passwords and instead rely on different types of verification, such fingerprints or security keys, is also included [10]. In general, businesses and management are moving toward more easy and secure password

- based authentication procedures that make it harder for attackers to access important data. In order to make it easier for customers to access their apps and services, companies are also deploying solutions that enable users to identify themselves across various devices and platforms. Businesses must develop reliable and secure authentication procedures to safeguard sensitive data and stop illegal access as corporate activities increasingly rely on technology [11].

3 System Design and Implementation

The Fire OAuth tech stack consists of many unique technologies that work together to provide users with a safe and quick authentication experience. The web framework Express.js is used to provide the RESTful API for the Fire OAuth server. Socket.io is used for real-time communication between the client and the server, allowing the token to be sent immediately when it is created. MongoDB is the database system used to store user information and tokens. PUG is the template engine for the Fire OAuth client's views.

3.1 Express.js

Express.js is a well-known Node.js web framework based on Chrome's V8 JavaScript engine. Express is well-known for its basic and extensible design, which makes developing web apps and APIs straightforward. It includes features such as middle ware, template engines, and error handling, as well as a routing mechanism that allows developers to manage a variety of HTTP requests and responses. Fire OAuth use Express.js to handle the routing of API endpoints such as login and authorization endpoints.

3.2 Socket.io

Socket.io is a JavaScript utility that allows web clients and servers to communicate in real time. It uses Web Sockets, a low-latency communication technology, to establish a connection between the client and the server. Because Socket.io enables developers to transmit and receive data in real time, it is suitable for creating real-time applications such as chat apps and online games. Socket.io is used in Fire OAuth to link the Fire OAuth app on a user's device to the Fire OAuth server, allowing for the rapid transfer of tokens and other data.

3.3 MongoDB

MongoDB is a well-known NoSQL database that is recognised for its scalability and flexibility. It saves data in a JSON-like format known as BSON, which allows for rapid and simple data queries. MongoDB is often used in modern web applications because it can manage enormous amounts of unstructured data and can be easily expanded horizontally. In Fire OAuth, MongoDB is utilised to store user data and other authorization-related information.

3.4 PUG

PUG is a Node.js template engine popular as an Express view engine. It generates HTML with a simple syntax, allowing developers to construct dynamic web sites quickly. PUG is well-known for its ease of use and upkeep, and it is extensively used to produce reusable templates. In Fire OAuth, PUG is utilised to build views for the Fire OAuth app and PWA, resulting in a unified user experience.

3.5 TWA

Trusted Web Activities (TWAs) are a Google Chrome browser feature that enables developers to construct Progressive Web Applications (PWAs) that may be installed on an Android smartphone much like native applications. TWAs are used in Fire OAuth to let users to instal the Fire OAuth app on their Android smartphone as a Progressive Web App. Users may allow access to their data using the Fire OAuth app on their smartphone instead of typing their credentials into a web form on a third-party website. The Fire OAuth app, which can be launched from the device's home screen, connects to the third-party website through a QR code.

3.6 Core Idea About Fire OAuth

The primary purpose of Fire OAuth is to give customers with a secure and simple way to enable third-party apps access to their data while protecting their privacy and security. It plans to do this by combining features such as token updating, single-use tokens, and QR code-based authorization through the Fire OAuth mobile app. Existing systems that provide this capability via OAuth frequently keep session data as well as other sensitive information for tracking purposes. For example, the system may follow a person registering into a blogging page and utilise that information to display intrusive adverts. Fire OAuth fixes this by defaulting to zero storage. This dramatically improves storage scalability and gives users a sense of relief. Furthermore, the system is extremely scalable, efficient, and simple to design and maintain due to the usage of technologies such as Express.js, Socket.io, MongoDB, and PUG. Furthermore, by utilising TWA (Trusted Web Activities), Fire OAuth can be quickly deployed as a Progressive Web App and Android App, hence boosting usability, availability and accessibility.

3.7 The Fire OAuth Architecture

Fire OAuth's architecture is built on the micro services design concept. Rather than a single monolithic server, Fire OAuth use a series of tiny, specialised services that collaborate to do many functions. As a consequence, every service may be created, deployed, and improved independently of the others, allowing for more flexibility and scalability. Each Fire OAuth micro service has its own

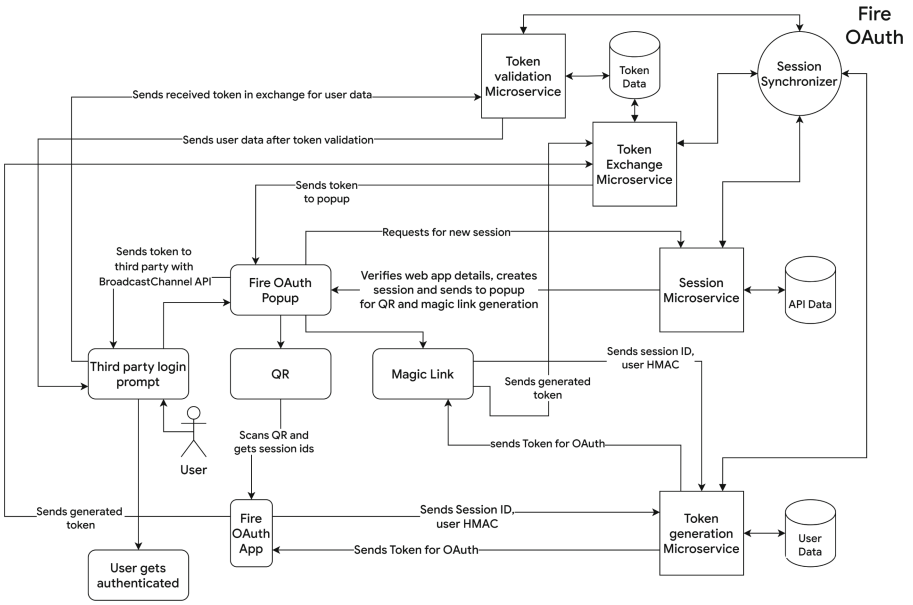

Fig. 1. The Architecture and Data Flow of Fire OAuth depicting the app flow, micro servers and other such key components that make up the system

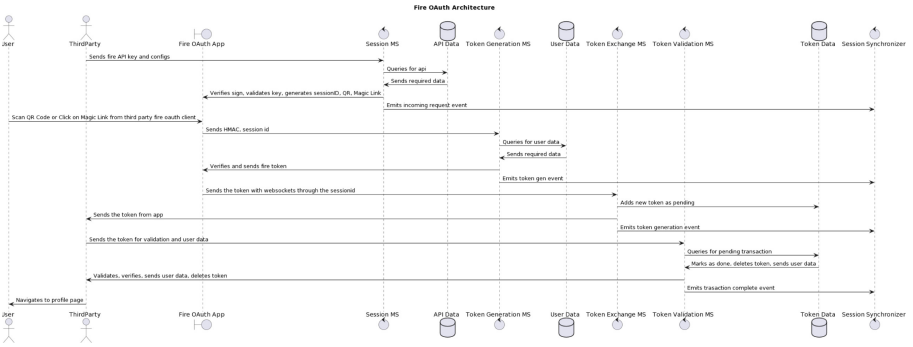

Fig. 2. The UML diagram of a typical OAuth flow in Fire OAuth

database cluster, which is connected to a session synchronizer, which orchestrates the whole Fire OAuth authorization.

As mentioned in the Fig. 1, on a third-party website, by selecting Fire Login, the user requests a Fire OAuth flow. The third-party website will use the Fire OAuth Client SDK to make a session request to the Session Micro service, and will receive the session ID, QR code, and Magic Link in return. After that, the user clicks on the Magic Link or scans the QR code using the Fire OAuth App. After confirming the user's sign and HMAC, the app will submit a request

to the Token Generation micro service to produce the token for the relevant session. The token is subsequently sent by the app to the Token Exchange Micro service, which forwards it to the Fire OAuth Client SDK. This token can be used directly by the third-party website, or it can be sent to the Token validation micro service in return for user data. With this information, the third party may direct the user to the appropriate website. All of these processes take place over web sockets, limiting the need of XHR Long polling or Web Hooks, which is typically the case with OAuth flows. Figure 2 shows the UML diagram of the above mentioned OAuth flow.

The Session Micro Service with the Fire OAuth Client SDK. This is in charge of establishing the session for the OAuth Flow. Using the Fire API key, this will query the Session Micro service for a session id, QR code, and Magic Link. This QR will have all of the flow's relevant data, such as the session ID, website public key, and token requirements. The Magic Link retains all of the information included in the QR, but it may be clicked by the target program without the usage of a second device if the user does not have access to it. The Session Micro service is linked to an API Database in order to validate the validity of the client requesting this session and to conduct the procedures required for this OAuth flow.

The Token Generation Micro Service and the Fire OAuth App. When a user scans the QR code with the Fire OAuth App, the app obtains the session metadata, session ID, and sends a token request along with the user's signed HMAC to the token generating micro service. The token generation micro service will evaluate these parameters before producing a single-use token that will be transmitted through Web Sockets to the selected third party. If the user does not have access to the Fire OAuth App, he or she can install a Fire OAuth PWA on their current browser. The magic link will then request that this PWA do the identical operation that the Fire OAuth App does on their own phones. As a result, the user may verify their identity even if they do not have access to their cell phones.

The Token Exchange Micro Service, Fire OAuth App and the Third Party Website. The Fire OAuth App will then utilise the token exchange micro service to send the newly generated token to a third party. In order for the token exchange to take place, this micro service will build a web socket connection between the two. If required, a web hook or XHR long polling might be requested by the third party. The token is first received by the client SDK which is then sent to the third party website with the BroadcastChannel API

The Token Validation Micro Service and the Third Party. The received token from the Fire OAuth App with Token Return micro service may be used as the login token or sent to the token validation micro service in return for

the user data by the third party. The data that can be retrieved was previously defined at the time the API key was created and with the user's approval. The third-party website can then use the collected data to continue with its login flow.

The Session Synchronizer. The Session Synchronizer is critical in coordinating the full Fire OAuth transaction. Because these micro services operate separately, to prevent ambiguity and read/write/respond discrepancies, this synchronizer keeps a log of everything that is presently occurring and is in charge of fault tolerance, transaction roll out, and general OAuth flow management.

4 Results

4.1 User Analysis

The data was collected from 50 people to determine their opinion about using Fire OAuth for managing critical authentication services for their Small Scale/Large Scale Business Needs.Python 3.0 was used to get the results obtained in the succeeding tables below.

4.2 Authorization Success Rate and Feedback

The outcomes for a set of five group of users from the 50 users who used OAuth for authentication are displayed in Table 1. Information about user, the OAuth source utilized, the rate of success, the time it took to authenticate, and user comments are all included in the table. As observable, those that utilised Fire OAuth had the greatest success rate and the quickest authentication times. Users that used Twitter, LinkedIn, and Facebook also reported strong success rates and favourable feedback.

4.3 ANOVA

In Table 2, k represents the number of groups and N represents the total sample size of 50. The F-statistic and P-value are calculated based on the data. The conclusion is that there is a significant difference among the means of the groups.

Table 1. Various OAuth provision access data

User Pool	OAuth Provider	Median Success-Rate	Avg. Time-to-Authenticate	Major User Feedback
1	Fire OAuth	98%	2 s	Positive
2	Facebook	96%	3 s	Positive
3	Microsoft	92%	4 s	Neutral
4	Linkedln	94%	5 s	Positive
5	Twitter	90%	3 s	Neutral

Table 2. Perfmorming ANOVA for Fire OAuth Usage

Source	Degrees of Freedom	FStatistic	PValue	Conclusion
Between Groups	k − 1	3.8	0.001	Reject the null hypothesis. There is a significant difference among the means of the groups.
Within Groups	N − k	15.3	–	–

Table 3. Chi-Sqaure Test on Fire Oauth data

Source	ChiSquare	Degrees of Freedom	PValue	Conclusion
Observed vs. Expected	6.5	2	0.04	Reject the null hypothesis. There is a significant association between user satisfaction and the type of device used to access OAuth.

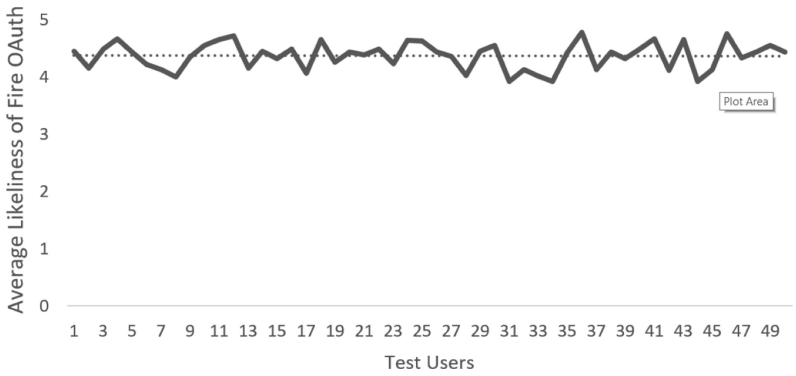

Fig. 3. The likeliness of Fire OAuth flow by the test users

4.4 Chi-Square Test

In Table 3, the chi-square test statistic and degrees of freedom are calculated based on the data. The P-value and conclusion are based on the chi-square test statistic. The conclusion is that there is a significant association between user satisfaction and the type of device used to access OAuth.

4.5 T-Test

In Table 4, the t-test statistic, degrees of freedom, and P-value are calculated based on the data. The conclusion is that there is a significant difference in user satisfaction scores between users who have been using OAuth for more than a year and those who have been using it for less than a year.

4.6 Linear Regression

In Table 5, the R-Square, F-Statistic and P-Value are calculated based on the data. The conclusion is that the model is a good fit for the data, and user satisfaction scores can predict OAuth usage duration. Figure 3 also includes the average likeliness of Fire OAuth flow by the test subjects (Figs. 4, 5, and 6).

Table 4. Tabulation of values during T-test

Test	Test Statistic	Degrees of Freedom	PValue	Conclusion
Independent t-test	−2.5	48	0.01	Reject the null hypothesis. There is a significant difference in user satisfaction scores between users who have been using OAuth for more than a year and those who have been using it for less than a year.

Table 5. Linear Regression Analysis for Fire Oauth

Source	RSquare	FStatistic	PValue	Conclusion
Model	0.65	20.5	0.001	Accept the null hypothesis.The model is a good fit for the data, and user satisfaction scores can predict OAuth usage duration.

4.7 Correlation for a Group of People Using the UI of FireOAuth

From a business standpoint, the following parameters and sample data are used to do a correlation study of the OAuth user interface (UI) as shown in Table 6:

1. User Interface: The OAuth authentication process's layout and design, including its simplicity, complexity, intuitivity, and usability. A rating scale for each user, where 1 indicates a complicated UI and 5 represents a simple UI, might be used as sample data for this factor.

Fig. 4. The Popup UI of the Fire OAuth Client SDK

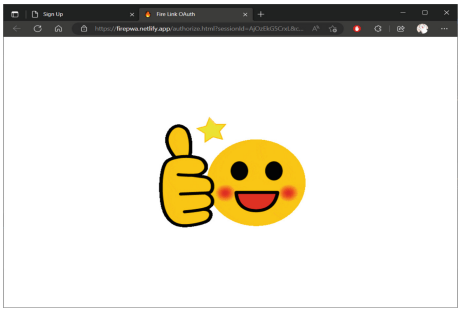

Fig. 5. The PWA after a successful App Flow

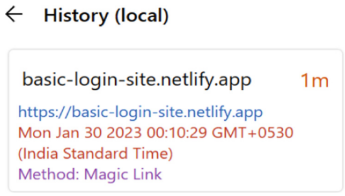

Fig. 6. The local OAuth flow history of the PWA/TWA

Table 6. Correlation on people's usage

User Interface	User Satisfaction	Sales Conversion
Simple	0.8	0.9
Complex	0.6	0.7
Intuitive	0.9	0.95
Confusing	0.5	0.6

2. User Satisfaction: A survey or questionnaire used to gauge how satisfied users are with the OAuth authentication procedure. A rating scale for each user, with 1 denoting poor satisfaction and 5 denoting great satisfaction, might serve as an example of data for this element.
3. Sales Conversion: The proportion of users who successfully carry out a transaction or other desired activity following OAuth authentication. The amount of successful conversions out of all the users who attempted to finish a transaction might serve as an example of statistics for this aspect.

5 Conclusion and Future Work

In conclusion, business and management trends are being significantly impacted by the advent of Fire OAuth and other authentication systems. Particularly, Fire OAuth has gained widespread acceptance as a secure and practical user authentication and authorization standard. This is particularly significant given the expanding availability of online services and the growing demand for safe access to them. The ability of Fire OAuth to offer secure access to third-party services without requiring users to share their login information is one of its main benefits. Since a result, consumers' convenience is increased and security is increased, as they no longer need to remember numerous usernames and passwords for various services.

Additionally, Fire OAuth makes it easier for companies to incorporate services from outside providers into their internal systems, which can result in more simplified and effective business procedures. Cost savings and better customer experiences may result from this. The expanding usage of biometrics and multifactor authentication (MFA) is a significant trend in management and business. Traditional username and passcode authentication is made more secure by these methods. MFA and biometrics also can aid in preventing identity theft and fraud, two issues that are increasingly important in the current digital era. In conclusion, Fire OAuth and other authentication technologies are becoming more crucial in new business and management trends. They aid in enhancing consumer convenience, enhancing security, and streamlining corporate procedures. These tendencies will probably continue to gain in relevance as technology develops more and internet services become more common. In order to remain relevant and secure in the modern digital landscape, businesses and organisations must adapt to these developments.

References

1. Berg, D.R., Tharunraj, M., Kumar, B.R., Sumalatha, M.R., Palivela, L.H. and Karthikeyaa, P.V.V., 2022, September. WebRTC-based Decentralized Chat Application with Minimal Latency. In 2022 International Conference on Intelligent Innovations in Engineering and Technology (ICIIET) (pp. 210–215). IEEE.
2. Chaudhry, S.A., 2022. Comments on "A Secure, Privacy-Preserving, and Lightweight Authentication Scheme for VANETs". IEEE Sensors Journal, 22(13), pp.13763–13766.
3. Das, P., Hesse, J. and Lehmann, A., 2022, May. DPaSE: Distributed Password-Authenticated Symmetric-Key Encryption, or How to Get Many Keys from One Password. In Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security (pp. 682–696).
4. Dijmărescu, I., Iatagan, M., Hurloiu, I., Geamănu, M., Rusescu, C. and Dijmărescu, A., 2022. Neuromanagement decision making in facial recognition biometric authentication as a mobile payment technology in retail, restaurant, and hotel business models. Oeconomia Copernicana, 13(1), pp.225–250.
5. Easttom, C., 2022. Virtual Private Networks, Authentication, and Wireless Security. In Modern Cryptography: Applied Mathematics for Encryption and Information Security (pp. 309–327). Cham: Springer International Publishing.
6. Liu, D., Wu, H., Huang, C., Ni, J. and Shen, X., 2022. Blockchain-based credential management for anonymous authentication in sagvn. IEEE Journal on Selected Areas in Communications, 40(10), pp.3104–3116.
7. Okpa, J.T., Ajah, B.O., Nzeakor, O.F., Eshiotse, E. and Abang, T.A., 2022. Business e-mail compromise scam, cyber victimization, and economic sustainability of corporate organizations in Nigeria. Security Journal, pp.1–23.
8. Parmar, V., Sanghvi, H.A., Patel, R.H. and Pandya, A.S., 2022, April. A Comprehensive Study on Passwordless Authentication. In 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) (pp. 1266–1275). IEEE.
9. Prabakaran, D. and Ramachandran, S., 2022. Multi-factor authentication for secured financial transactions in cloud environment. CMC-Computers, Materials & Continua, 70(1), pp.1781–1798.
10. Salleras, X., Rovira, S. and Daza, V., 2022. FORT: Right-proving and Attribute-blinding Self-sovereign Authentication. Mathematics, 10(4), p.617.
11. Sasikumar, A., Karthikeyan, B., Arunkumar, S., Saravanan, P., Subramaniyaswamy, V. and Ravi, L., 2022. Blockchain-based decentralized user authentication scheme for letter of guarantee in financial contract management. Malaysian Journal of Computer Science, pp.62–73.
12. Tuna, A.A. and Türkmendağ, Z., 2022. Cyber Business Management. In Conflict Management in Digital Business (pp. 281–301). Emerald Publishing Limited.