INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 5, May 2026 | ISSN: 2456-4184 | IJNRD.ORG

Next-Generation Authentication: Combining Blockchain and Two-Factor Security

1st B. Jaanakki Devi
Department Of Advanced Computing And Analytics
Vels Institute Of Science, Technology &
Advanced Studies Chennai, Tamil Nadu
janakibaskaran65@gmail.com

2nd Ms. R. Jeya Shree, Assistant Professor
Department Of Advanced Computing And Analytics
Vels Institute Of Science, Technology &
Advanced Studies Chennai, Tamil Nadu
rjeyashree.scs@vistas.ac.in

Abstract - In the contemporary digital era, social as well as business applications are getting more advanced and this has resulted in a heightened need to verify identities in a stronger manner. Two-Factor Authentication (2FA) is an essential component in the improvement of the old system of authentication, though it is not without drawbacks. Even with effective encryption systems, conventional 2FA techniques remain vulnerable to security attacks of SIM swapping, Denial-of-Service (DoS), and Man-in-the-middle (MITM) attacks. Moreover, such systems are highly centralized, which is also a considerable threat because one breach will enable attackers to depersonalize legitimate users or discontinue the whole service. In an effort to deal with such issues, this paper suggests a decentralized solution to 2FA. The system will have enhanced security by distributing the authentication process and the use of key cryptography that is publicly available. Under this model, the user secrets are encrypted with the public key of the wallet of the user so that they have full control over their authentication credentials. Moreover, even when they are not connecting to their main device, they are able to create special passwords on their own. Since the need of more robust digital security has been growing steadily, the shift to decentralized approaches to 2FA in favor of traditional centralized one will ensure more security and leave the user with more freedom.

INTRODUCTION

In today's digital world, digital identity is vital to ensure that both business and personal services are accessed in a safe manner and provide non-repudiation. Two-Factor Authentication (2FA), which has become common and commonly practiced, allows organizations to better manage their identity verification processes in applications. Essentially, two-step verification (2SV) or dual-factor authentication (DFA) sends customers two different means of verifying their identity when logging on. The extra security provided by the second verification factor makes it difficult for attackers to gain access to a user's device or online account. Even though the user's password may have been compromised, there isn't enough information for an attacker to gain access to the authentication process on the device or online account. The additional level of security provided by 2FA makes it much more secure than Single-Factor Authentication (SFA), which relies solely on a password, with respect to the security of user credentials included in the authentication process as well as user resources.

2FA is usually made up a combination of a password (the first factor) and an OTP (the second factor). OTPs are used in conjunction with standard username/password pairs for authentication purposes. However, the ability to generate and control OTPs can also present issues for system security. One popular OTP algorithm is TOTP. TOTP provides an OTP that will continue to change over time, usually at intervals of 30 seconds. To calculate TOTP, you multiply the current date/time with a "shared secret" that both the user and the server have agreed upon prior to authentication. This establishes a synchronized point in time from which both the user and the server can generate matching OTPs. The synchronization of date/time when generating TOTP will provide a secure, repeatable method of authenticating a user's identity due to the time-sensitive nature of TOTP.

Typically, a 2FA login will require both a password (the first factor) and a one-time password (OTP) (the second factor), to authenticate a user. OTPs can be used along with standard username/password authentication mechanisms. Creating and managing OTPs can pose challenges to system security. An example of an OTP generation algorithm is TOTP, which provides an OTP that will be variable over time based on a defined time window, frequently 30 seconds. TOTP can be computed by taking the current date/time and multiplying it by a "shared secret," which the user and server have established before the user attempts to authenticate. The user and server will be able to create matching OTPs, as they will share an equal amount of time to create them. Because the generation of the OTP is time-dependent, TOTP will create a secure and repeatable method of validating an individual's identity.

LITERATURE SURVEY

Study 1: The article is called A Microservices and Blockchain-Based One-Time Password (MBB-OTP) Protocol for Security-Enhanced Authentication and suggests the application of a decentralized two-factor system of authentication to enhance protection against Man-in-the-Middle (MITM) attacks. This is in contrast to the traditional 2FA systems that make use of centralized elements, since the effect is that there is no single point of failure in the entire system, and the system is still available even when one of the elements is breached. Also, smart contracts are applied to ensure the integrity of data, non-repudiation, and immutability, i.e. the data stored cannot be changed or removed. The system is useful in countering attacks like Denial-of-Service (DoS) and MITM. Nonetheless, it also relies on the OTP generation over SMS, which creates a point of weakness to the SIM swapping attacks.

Study 2: The article is called Two Chain: Leveraging Blockchain and Smart Contracts to Two-Factor Authentication and presents a blockchain-based 2FA system, in which it belongs to something you have. In this system, the users have to demonstrate the ownership of a private key

associated with an already registered public key on the server. This check is done following the first authentication phase in which the user is required to give his / her username and password. Through the use of blockchain and cryptographic keys, the system will increase the security of authentication processes and minimize the use of the conventional OTP systems.

Study 3: The other study on the topic of Trustless Two-Factor Authentication using Smart Contracts in Blockchains offers a decentralized 2FA model, in which users generate authentication tokens by themselves without reliance on third-party services, like SMS or email. The system uses Command Line Interface (CLI), which is not easy to use, but it eliminates SIM-jacking attacks and enhances the overall security. The design is based on blockchain and is therefore transparent and audit-able. Nonetheless, the update of smart contracts needs changes in all deployed Pluggable Authentication Modules (PAM), and network interruptions can have an impact on the accessibility of the system, especially SSH connections.

Study 4: The article Two-Factor Dynamic Identity Authentication Scheme for Data Trading Based on Alliance Chains is devoted to the theme of the secure authentication in data trading systems. Although the proposed scheme is based on the solution of various limitations of the traditional identity verification methods, it nonetheless relies on the Unique Identity Token (UIT) certificates, which can prove hard to manage as the network expands. There is also the likelihood that the current storage and retrieval systems could not scale effectively in large alliance chain systems. The model is not as widespread in its applicability to real-life authentication systems as its use in data trading situations because this model is still at its initial phases and is mostly designed as a data trading tool.

Study 5: The research article by the title of SMS OTP Security (SOS): Hardening SMS-Based Two-Factor Authentication is concerned with enhancing the security of OTP delivery via SMS. Even though it presents more powerful protection measures, the solution has a number of shortcomings. It involves adjustments to default messaging applications, which causes device and platform compatibility problems. Cryptographic operations lack transparency as well, which can become a concern among the users. Moreover, its success in countering sophisticated attack methods is questionable. Even with improvements, SMS might not be secure enough to be used in high-security settings where more secure authentication like biometrics and hardware tokens are required. Also, the system fails to deal with weaknesses in other communication systems such as email, thus making the system ineffective.

PROPOSED METHODOLOGY

Next-Generation Two-Factor Authentication (2FA) proposed is aimed at enhancing the level of security and is based on decentralized storage and sophisticated encryption measures. This method is a decentralized method of storing 2FA secrets in a decentralized setting under the use of public-key cryptography in contrast to the traditional 2FA which relied on centralized servers and third-party services. This guarantees the user full ownership and control of his authentication data. Under this system, the 2FA secrets are encrypted using the public key of the user that is linked to his or her blockchain wallet. Consequently, these secrets can only be decrypted and accessed by the corresponding key holder who happens to be the user. To enhance the security further, the system uses AES-GCM encryption algorithm, which offers confidentiality and integrity of data. Such a hybrid of public-key encryption and AES-GCM has a great potential of mitigating the vulnerabilities that are prevalent in more traditional 2FA models, including SIM swapping and Denial-of-Service (DoS) attacks.

The system also makes use of the decentralized storage that is supported by the Zero-Knowledge (ZK) Rollup protocols. This provides an efficient, secure and scalable storage of encrypted authentication information and user privacy is maintained. The proposed model does not need centralized authorities and outside communication channels (e.g. SMS or email), hence the absence of most attack vectors in the system and increased system resiliency. The main strength of this solution is that the users can create the dynamic One-Time Password (OTP) on their own and anytime without relying on a primary device or an external service. The system is also compatible with Web2 and Web3 applications hence it is flexible and can be used extensively across the platforms.
In general, the suggested mechanism provides secure, decentralized and user-friendly authentication framework. It is more secure, offers better user controls, and is more flexible than the conventional 2FA systems.

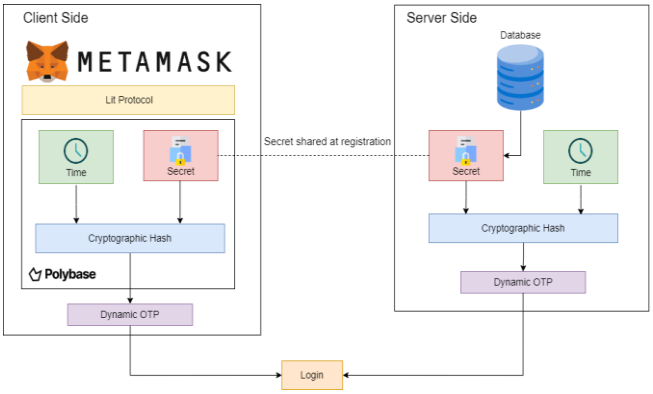


Fig.1 Architecture of Next-Gen ZK

Work Flow

Step 1: Customers can use any EVM-compatible wallet to access the application with any device.

Step 2: The system generates an encrypted secret of the user based on the 2FA secret and the public key of the wallet address of the user.

Step 3: The secret is encrypted and stored in decentralized storage in the company of Zero-Knowledge (ZK) Rollup technology.

Step 4: In the cases where authentication must be done, the stored secret is retrieved and decrypted with the help of the private key in the possession of the user.

Step 5: Dynamic One-Time Passwords (OTPs) are created through the decrypted secret.

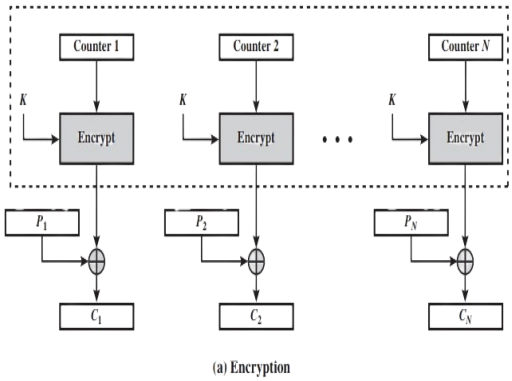
Step 6: OTP created is the second authentication factor and it may be utilized on a Web2 and Web3 platform.

ARCHITECTURE DIAGRAM

AES-GCM (Advanced Encryption Standard - Galois/Counter Mode) algorithm is applied in the proposed Next-Gen ZK system in order to store user 2FA secrets securely. This type of encryption is a favourable choice over AES-ECB (Electronic Codebook) and AES-CBC (Cipher Block Chaining) because it is a mode of encryption that has better security and efficient processing. AES-GCM provides authenticated encryption that does not only provide data confidentiality but also data integrity. It creates an authentication tag and the encrypted data to enable the system of the system to check whether the data is not changed or interfered with. It is especially relevant to the protection of 2FA secrets and related data, including account information. AES-GCM will ensure the overall credibility of the authentication system by avoiding unauthorized changes. In addition, AES-GCM allows only users with the proper keys of cryptography to decrypt and retrieve sensitive data. This makes it much harder to be hacked and makes the Next-Gen ZK system of authentication much stronger.

Encryption:

1. This system first receives the wallet public key of the user.
2. Each encryption is made random and secure by generation of a special Initialization Vector (IV).
3. The AES-GCM algorithm is used to encrypt the 2FA secret key with the involvement of the public key and an IV generated.
4. The encrypted data is then stored in a decentralized storage which is secure.

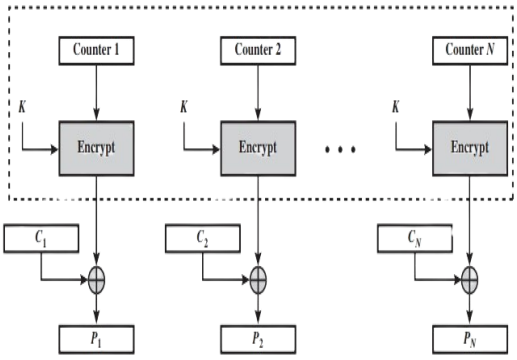


(a) Encryption

Fig.2 Advanced Encryption Standard Galois Counter Mode (AES GCM) Encryption Process

Decryption:

1. The 2FA secret, which is encrypted, is stored in a decentralized storage.
2. The system deciphers the information with the wallet private key of the user.
3. A dynamic 6-digit One-Time Password (OTP) is generated using the TOTP algorithm using the decrypted secret.
4. This OTP is then created as the second factor of authentication to gain access to applications on both Web2 and Web3 platforms.



(b) Decryption

Fig.3 Advanced Encryption Standard Galois Counter Mode (AES GCM) Decryption Process

VARIOUS PHASES / METHODOLOGIES

There are also numerous applications of Two-Factor Authentication (2FA) that are currently popular such as Google Authenticator, Authenticator by Authy, and Microsoft Authenticator. According to their preferences, users are in a position to choose an application which will best suit the services they are planning to obtain. They are available in official websites like the Apple App Store or the Google Play Store. Once it is installed, the user should set up the 2FA program as per the guidelines. It is normally done by scanning a QR code that is generated by the service provider. In the QR code, there is a secret key that allows the application to use the TOTP protocol to generate time-based One-Time Passwords (OTPs). In order to recover the accounts in the event of a loss of the device or the deletion of the application, the users are usually given backup codes that must be safely stored.

In order to use 2FA, customers have to go to security settings of their accounts and enable it. At the time of set up, the application is connected to the account by scanning a QR code or filling a generated key manually. The system can then prompt a verification OTP to ascertain successful configuration. When this feature is enabled users are required to fill their password and a time-bound OTP that is issued by the application with every attempt of logging in. Despite the fact that 2FA dramatically improves the security of accounts, it is not entirely resistant to attacks. Unauthorized access may still be caused by risks like malware, phishing attacks, and device compromise. Thus, the best practices that users are advised to use are to ensure that applications and devices are up to date, use strong and unique passwords (better use password manager), and activate device-level security, including encryption and biometric authentication. As well, the users are advised to regularly update their secured accounts and eliminate idle services. Another thing is ensuring that the authentication applications are downloaded only one can be sure to avoid malicious or fake applications.

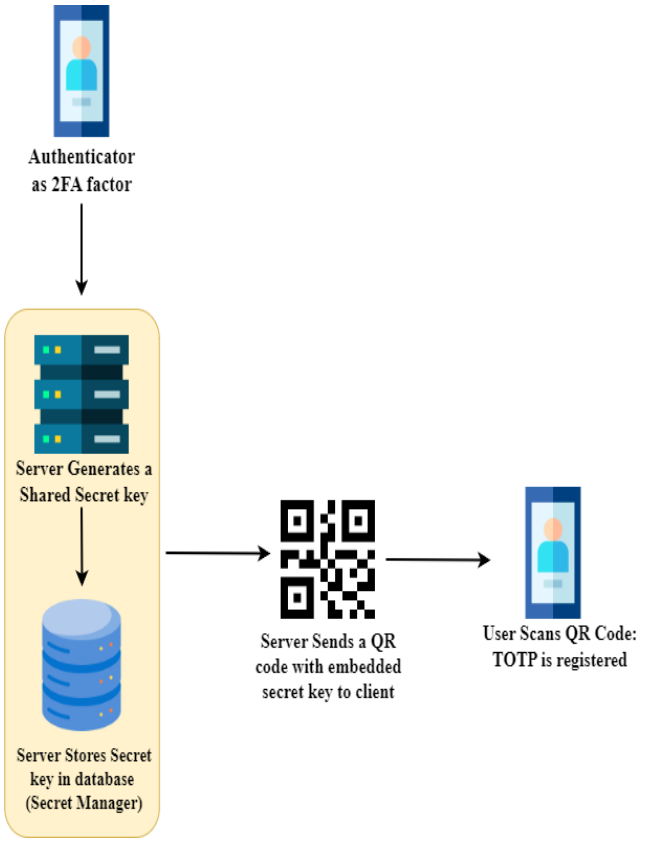


Fig.4 Here's a simplified flow when TOTP authenticator apps are registered

Step 1: After authentication, the user is required to input their username and password. Then, they make 2FA active, and choose an authenticator application as the second authentication factor.

Step 2: The client obtains a secret key (seed), and it is stored in a QR code or a URL. This seed is safely placed on the server to be checked in future.
Step 3: With the help of the authenticator application, the user scans the QR-code or opens the offered link. This app keeps the seed safely on the personal computer of the user and this registers the seed.

Step 4: A confirmation step is carried out on the verification by a TOTP code.

In the traditional systems, the whole process is managed in a centralized service, thus making it easier to attack. In case of compromising, attackers can access sensitive information, including OTPs.

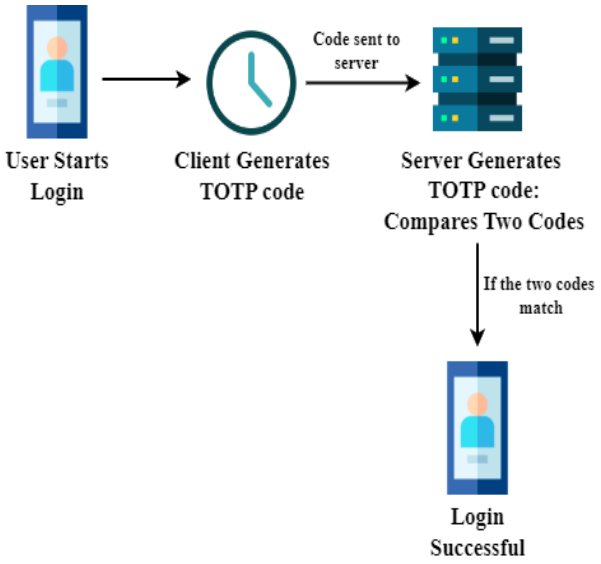


Fig.5 Here's a simplified flow showing TOTP

Step 1: First the user starts the authentication procedure by passing the initial authentication factor (username and password).

Step 2: With the seed that had been previously stored and the current time (moving factor), the client device computes a TOTP code and transmits it to the server.

Step 3: The server calculates a TOTP code independent of the server with the same seed and time factor. It then contrasts this code with the one obtained by the client.

Step 4: In case the two codes are identical, the user is authenticated and allowed to access.

This reorganised clarification shows the functionality as well as constrained nature of traditional 2FA frameworks, resulting in the significance of more secure and decentralised frameworks.

INPUT

A next-generation authentication system uses the combination of user identity, blockchain credentials, and two-factor verification data to create a secure and decentralized access control mechanism. The user first authenticates with Web3 wallet credentials with browser-based wallets (e.g., MetaMask or other Wallet Connect-compatible apps). The system substitutes the traditional username-password model, with the public wallet address of the user a unique identifier of blockchain kind. As a further identity verification, the user signs a cryptographical message with his or her own key. This signature is evidence of the signature of the wallet without revealing the private key. The signature is verified by the system to enable the authenticity of the user to be established. The system will also use Two-Factor Authentication (2FA) in addition to blockchain-based verification to offer an extra security measure. Time-Based One-Time Password (TOTP): An encryption was created on

the client side based on a secret key and time which the user has to input to be verified.

Other inputs that could be taken into consideration in the system to increase security include:

- Secured user information by means of decentralized encryption.
- Details of devices or browsers to validate a session.
- Access control or whitelisting IP address.
- Limitation of rate to prevent unauthorized repetitive attempts of logging in.

The processing of all the critical inputs is encrypted or cryptographically verified. Through combining the use of blockchain-based identity verification and 2FA, the system provides a very secure, tamper-resistant, and user-centric authentication system.

PSEUDOCODE

BEGIN
Authentication involves confirming the identity of the user.
Step 1: Authentication initiation: Authentication is a process of verifying the identity of the user.
Initiate the authentication process.
Step 2: Verify the Availability of Wallet.
Check the presence of a Web3 wallet
(e.g.,window.ethereum) in the browser of the user.
In case of no wallet, the user should install one and end the process.
step 3: Wallet Connection.
Connect WalletConnect using Web3Modal.
Secure the wallet of the user using wagmi and viem hooks.
When connected successfully, get wallet address of the user.
Step 4: Signature authentication of users.
Combine the Polybase and Lit Protocol with the WalletConnect instance.
Ask the user to sign cryptographic messages on:
Lit Protocol Authentication.
Record creation in Polybase
Sign authentication Verifies the signing of the wallet.
Step 5: Data Encryption
Accept user data as input.
Lit Protocol is used to encrypt the data to guarantee confidentiality and privacy.
Step 6: Decentralized Storage of Data.
Use write gating collections which are public so as to apply decentralized secure storage.
Part 7: Retrieval and Decryption Data.
Fetch encrypted data when needed Polybase.
Authorized access to the data uses Lit Protocol to decrypt.
Step 8: OTP Generation
Create a One-Time Password (OTP) at the client-side.
Combine the secret key with the present moment, Time-Based One-Time Password (TOTP) algorithm.
Step 9: OTP Verification
Ask the user to input the OTP that has been generated.
Compare system generated OTP with the entered OTP:
If matched - proceed further
If not matched - deny access
Step 10: Implement Security Controls.
Limit the rate to discourage brute force attacks.
IP whitelisting should be used to limit unauthorized access.
Track user actions constantly and keep audit logs in order to identify a suspicious activity.
Step 11: Deal with Multiple Signature Requests.
Consider injected WalletConnect with the aim of reducing the number of signature requests.
Nevertheless, several signatures can be maintained because of the version changes between WalletConnect (V1 to V2).
Step 12: Access Decision
ACL is allowed when authentication and OTP authentication are both successful.
Otherwise, deny access.
Step 13: Continuous Monitoring
Carry out constant analysis of the system on potential weaknesses.
Identify and act on security threats before.
END

OUTPUT

The result of the proposed next-generation authentication system will be the final result following the blockchain-based identity verification and two-factor authentication. After the user manages to provide a valid cryptographic signature and connect their Web3 wallet to the system, the system identifies the owner of the wallet and verifies the user. After that, the One-Time Password (OTP) code, which is created based on Time-Based One-Time Password (TOTP) algorithm is authenticated. In case of a successful verification of both steps, the user is given secure access.
The major deliverables of the system are:

- **Authentication Status:** The status of a user is whether the user is authenticated successfully or the user is denied access based on the results of the verification.
- **Session / Access Token:** A secure data is created that is used to hold on to the users session to allow further interaction with it without repeating the authentication process.
- **Access to Decrypted Data:** Authorized users are able to safely access and decrypt their data stored in decentralized storage systems.
- **Authentication of Actions:** Verified users can only execute operations like storing records, accessing services, or inter-relating with blockchain-based applications.
- **Audit Logs and Activity Tracking:** The system logs all attempts to log-in, successful log-in attempts and unsuccessful attempts to log-in to monitor and analyze security.
- **Security Alerts:** An alert should be issued whenever a suspicious activity happens and relevant restrictions can be imposed in case of a failed attempt to log-in.

In the event of authentication failure at any point, either through an invalid signature or wrong OTP, the system denies it and gives appropriate error messages. Further attempts to prevent abuse may also be blocked on a temporary basis by security measures like rate limiting. On the whole, the system provides the highest rates of security, privacy, and trust as

only verified and authorized users are allowed to access the resources.

RESULTS AND DISUCSSION

The Next-Gen-ZK system overcomes the shortcomings of the traditional two factor authentication scheme like use of SMS, email and regular authenticator apps. It supports user 2FA secrets, which are safely stored using decentralized storage and public-key encryption, and it supports the creation of dynamic One-Time Passwords (OTPs). These OTPs enable users to authenticate on both Web2 and Web3 systems with a wallet compatible with EVM. The authentication starts on the sign-in page, where the users enter the system with any device, which has an EVM wallet like a desktop browser or a mobile application. In contrast to the traditional system of 2FA where the user is required to use a particular registered device to obtain the OTPs, this system avoids the dependency on the device. In the event that a user loses his or her device, he or she can still gain access to it using his wallet credentials. Before signing in to the Next-Gen-ZK, users that do not possess a wallet, have to create one.

The site provides various wallets and offers users the opportunity to continue through familiar wallets like MetaMask, Coinbase Wallet, Rainbow, and others through WalletConnect SDK. Upon successful connection of their wallet, and successful authentication, the users will be redirected to the main interface. On the home page, the users are able to change and maintain their 2FA preferences. The system will enable a user to create and manage a number of accounts (and this is limitless to a certain number) each of which is a secret one. The wallet address is clearly displayed and the user can administer his accounts by adding and removing entries. To activate 2FA on a certain account, one will be required to enter information like the name of the service, the account number (e.g. email) and a special secret code. This secret key may be typed in or scanned by a QR-code, depending on the device one is using. The Next-Gen-ZK system provides better security as opposed to the traditional applications that use authenticators because all the keys in these applications are stored in plaintext.

In encryption, the system makes use of the Lit Protocol that uses the AES-GCM algorithm. This is a better form of encryption than the AES-CBC and AES-ECB that were earlier modes of encryption because they were known to have weaknesses. The data is thereafter encrypted and stored in Polybase which is a decentralized database so that sensitive data is not accessed by unauthorized individuals. The setup procedure involves having users submit a cryptographic signature to give permission to the encryption and storage of their 2FA secrets. These secrets are then encrypted and safely stored and accessible only by the authorized user. Once configured successfully, the system creates OTPs automatically after every 30 seconds based on TOTP algorithm. This process is done by combining the stored secret key with the current time stamp to generate time sensitive codes. The OTP generation is done on the local side of the client which reduces key exposure risks. To authenticate, the user will be required to enter the generated OTP and the system will verify the user. The same server generates a corresponding OTP using the same secret and time factor. In case of a match between the two OTPs, the user is authenticated and allowed to gain access.

SCREENSHOTS

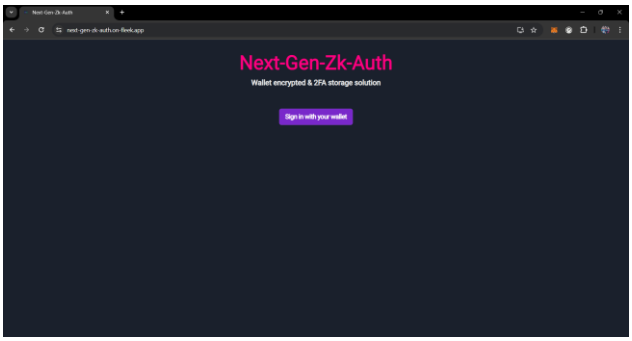


Fig.6 illustrates the sign-in page

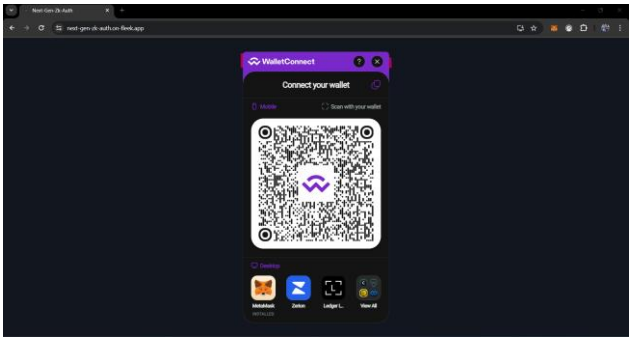


Fig.7 illustrates the wallet connect options

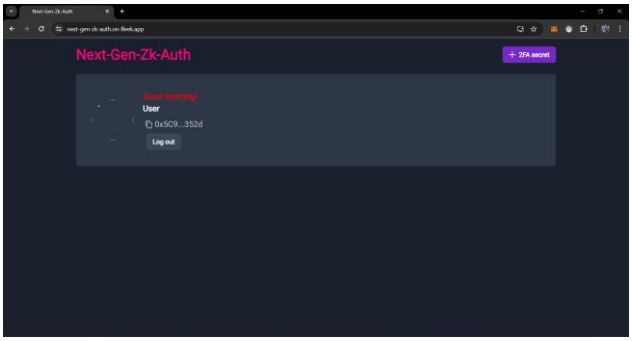


Fig.8 illustrates the landing screen

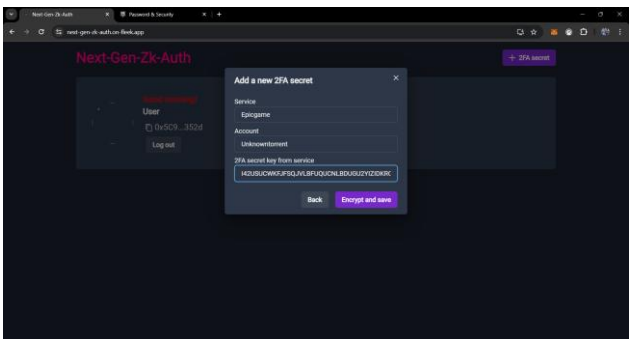


Fig.9 illustrates the adding two factor secrets

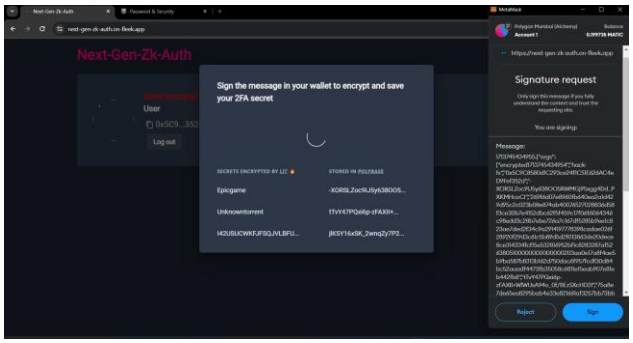


Fig.10 illustrates the signature signing to adding two factor secrets

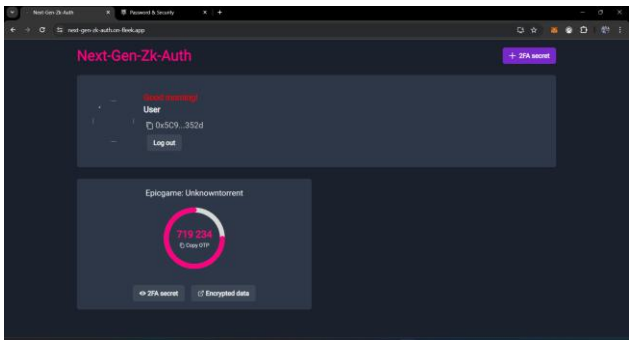


Fig.11 After adding the two factor secrets

CONCLUSION

Suggested Next-Gen-ZK system suggests a decentralized system of Two-Factor Authentication (2FA), overcoming the weaknesses of traditional concept of authentication. With the help of decentralized storage and public key encryption, user 2FA secrets are stored safely and without use of any central authority or trusted third party. This architecture ensures security besides the absence of a common dependency, like SMS or email-based verification systems.

This allow users to autonomously create One -Time Passwords (OTPs), allowing a seamless Web2 and Web3 authentication process through wallet details. The proposed model facilitates privacy of the users and also gives acces over multiple device, which is unlimited to the device specific authentication. Commonly, the given decentralized 2FA architecture shows a safe, trustworthy, and user-friendly framework which integrates high level security of secrets in authentication and enhanced usage. It is a successful solution, which has high scalability and practical in ensuring digital identities which is secure in the ongoing applications, which gives strong alternatives to the current means of authentication.

REFERENCES

[1] Alessio Catalfamo, Armando Ruggeri , Antonio Celesti, Maria Fazio, and Massimo Villari, "A Microservices and Blockchain Based One Time Password (MBB-OTP) Protocol for Security-Enhanced Authentication" IEEE 2021.

[2] Yustus Eko Oktian, Sang-Gon Lee and Hoon-Jae Lee,"TwoChain: Leveraging Blockchain and Smart Contract for Two Factor Authentication" ,IEEE 2020.

[3] Varun Amrutiya, Siddhant Jhamb, Pranjal Priyadarshi, Ashutosh Bhatia, "Trustless Two-Factor Authentication using Smart Contracts in blockchains", IEEE 2019.

[4] Bin Zhao,Wenyin Zhang,Yilong Gao,Fengmei Chen, "Two-factor dynamic identity authentication scheme for data trading based on alliance chains",Springer 2023.

[5] Christian Peeters, Christopher Patton, Imani N. S. Munyaka, Daniel Olszewski, Thomas Shrimpton and Patrick Traynor, "SMS OTP Security (SOS): Hardening SMS-Based Two Factor Authentication", ASIA CCS 2022.

[6] Mauli Bayu Segoro and Prasetyo Adi Wibowo Putro, "Implementation of Two Factor Authentication (2FA) and Hybrid Encryption to Reduce the Impact of Account Theft on Android-Based Instant Messaging (IM) Applications", IEEE 2021.

[7] Ahmed Tanvir Mahdad, Mohammed Jubur and Nitesh Saxena,"Breaking Mobile Notification-based Authentication with Concurrent Attacks Outside of Mobile Devices", ACM MobiCom '23 2023.

[8] Ivan Homoliak, Dominik Breitenbacher and Ondrej Hujnak,"SmartOTPs: An Air-Gapped 2-Factor Authentication for Smart-Contract Wallets", ACM Conference on Advances in Financial Technologies 2020.

[9] Xinming Yin, Junhui He , Yi Guo , Dezhi Han , Kuan-Ching Li ,and Arcangelo Castiglione, "An Efficient Two-Factor Authentication Scheme Based on the Merkle Tree", Sensors by MDPI 2020.

[10] Radhesh Krishnan Konoth, Bjorn Fischer, Wan Fokkink, Elias Athanasopoulos,Kaveh Razavi and Herbert Bos, "SecurePay: Strengthening Two-Factor Authentication for Arbitrary Transactions", IEEE European Symposium on Security and Privacy (EuroS&P) 2020.

[11] Vasilis Papaspirou, Leandros Maglaras, Ioanna Kantzavelou, Naghmeh Moradpoor and Sokratis Katsikas, "A Blockchain-based Two Factor Honeytoken Authentication System" arxiv 2023.

[12] Linsheng Yu, Mingxing He, Hongbin Liang, Ling Xiong and Yang Liu, "A Blockchain-Based Authentication and Authorization Scheme for Distributed Mobile Cloud Computing Services", Sensors by MDPI 2023.

[13] Duisen Gulsezim ,Seiitkaliyeva Zhansaya,Abdul Razaque,Yestayeva Ramina,Ahmed Oun,Raouf Ganda,Muder Almiani,Fathi Amsaad, "Two Factor Authentication using Twofish Encryption and Visual Cryptography Algorithms for Secure Data Communication" , IEEE 2019.

[14] Miqi Wu, Lin You , Gengran Hu , Liang Li, and Chengtang Cao, "A Blockchain-Based Hierarchical Authentication Scheme for Multi Server Architecture", Hindawi 2021.

[15] Ali Abdullah,Hosam Alamleh,Jean Gourd,Hatwib Mugasa "Zero Effort Indoor Two Factor Authentication", IEEE 2020.