A Novel Zero-Knowledge Proof-based Verification and Authentication Protocol (ZKPVA) for Web Security

Raj Mehta[1], Prasham Shah[2], and Pimal Khanpara[3]

Department of CSE, Nirma University, Ahmedabad, India
22bce183@nirmauni.ac.in[1] 22bce325@nirmauni.ac.in[2]
pimal.khanpara@nirmauni.ac.in[3]

Abstract. Zero-Knowledge Proofs (ZKPs) are a cutting-edge cryptographic technique that enables secure authentication in online applications—such as e-commerce platforms, online banking, social media, cloud storage, and SaaS platforms—without revealing sensitive information. This study explores the application of ZKP-based authentication methods to enhance web security, particularly in mitigating risks such as data breaches and identity theft. We present a comprehensive analysis of ZKP principles, critically evaluate existing approaches, and introduce a novel Zero-Knowledge Proof-based Verification and Authentication (ZKPVA) protocol specifically tailored for web applications. The workflow of the proposed framework is visually illustrated and compared with existing methods to emphasize its efficiency and security benefits. Our findings highlight the critical role of ZKP-based authentication in protecting sensitive data while meeting the growing demand for robust and efficient security solutions in modern web applications.

Keywords: Zero-Knowledge Proofs (ZKPs), Web Security, Authentication, Cryptography, Secure Communication

1 Introduction

In the digital era, web applications play a crucial role across diverse domains, including e-commerce, banking, healthcare, and social networking. These applications frequently process sensitive data, such as personal information, financial transactions, and authentication credentials, making them attractive targets for cyberattacks [Muzammil et al(2024)]. Ensuring user protection and maintaining trust require the implementation of secure authentication mechanisms [Zhuravel(2024)]. However, traditional methods, such as password-based systems, have inherent vulnerabilities that compromise their effectiveness, prompting the need for innovative, secure, and user-friendly solutions.

Authentication, the process of verifying the identity of a user, is fundamental to Web-to-Web security. Historically, this process has been based on passwords or shared secrets, where users prove their identity by providing information only to them [Veeramachaneni(2025)]. Although straightforward, this approach suffers

from critical weaknesses. Users often reuse simple and easily guessable passwords, making them susceptible to brute-force attacks [Sahoo et al(2024)]. Phishing attacks deceive users into divulging their credentials, while database breaches expose sensitive information, jeopardizing both users and systems [Birthriya et al(2025)].

To address these challenges, modern systems have incorporated advanced authentication techniques such as multi-factor authentication (MFA), biometric authentication, and public-key infrastructure (PKI). MFA enhances security by combining multiple verification factors, such as a password, a trusted device, or biometric data. Although effective, MFA often reduces user convenience and increases implementation complexity [Kantipudi et al(2024)]. Similarly, biometric authentication, while offering strong protection, raises significant privacy concerns, as compromised biometric data cannot be replaced [Alrawili et al(2024)]. PKI provides robust security through asymmetric cryptography, but requires meticulous management of cryptographic keys and digital certificates, adding operational challenges [Choudhary et al(2024)].

Despite advancements in authentication methods, many still struggle to balance security, usability, and computational overhead. This highlights the need for a secure, efficient, and privacy-preserving solution for web-application authentication [Jiang et al(2024)]. Web applications, often accessed over untrusted networks, face constant threats such as eavesdropping, session hijacking, and replay attacks [Hoffman(2024)]. These applications must accommodate a diverse user base, from individuals with limited technical expertise to enterprises handling sensitive operations. Authentication mechanisms must ensure confidentiality, integrity, and scalability while preserving user privacy by minimizing the exposure of sensitive data [Anyanwu et al(2024)].

Zero-Knowledge Proofs (ZKP) provide a solution to the limitations of traditional authentication methods by allowing one party (the Prover) to prove possession of a secret without revealing it [Zhou et al(2024)]. This feature ensures secure authentication while preserving privacy, making ZKP ideal for web applications. The proposed ZKPVA protocol, which includes four phases—Registration, Commitment, Challenge-Response, and Verification—employs advanced cryptography to protect sensitive information, such as private keys, and prevents its exposure during authentication. This approach mitigates common attack vectors like phishing and eavesdropping, significantly enhancing web application security.

2 Related Works

Zero-Knowledge Proofs (ZKPs) offer a powerful cryptographic method for secure, privacy-preserving authentication in online applications. This section reviews various ZKP-based authentication schemes, highlighting their strengths and limitations.

Wu et al. [Wu et al(1998)] introduced the Secure Remote Password (SRP) protocol, enhancing password security by preventing password transmission.

Table 1. Comparative Analysis of ZKP-Based Authentication Methods

Reference	**Domain**	**ZKP Integration**	**Strengths**	**Limitations**
[Wu et al(1998)]	Password authentication	No	Robust password-based security	Relies on static passwords
[Grzonkowski et al(2008)]	Web applications	Yes	Lightweight and efficient	Does not address advanced web threats
[Pathak et al(2021)]	Secure authentication	Yes	Mitigates replay and MitM attacks	Limited scalability
[Chistousov et al(2022)]	Adaptive protocols	Yes	Dynamically adapts to threats	May introduce latency
[Shahrouz and Analoui(2024)]	Vehicular ad-hoc networks	Yes	Privacy-preserving with blockchain	Computationally intensive
[Su et al(2024)]	IoT devices	Yes	Efficient for IoT device authentication	Limited application to web systems
[Jiang and Guo(2024)]	IoV	Yes	Anonymous authentication, privacy-focused	Complexity affects real-time performance
ZKPVA	Web applications	Yes	High security, efficient, mitigates breaches	Application-specific

However, its reliance on static passwords makes it vulnerable to phishing or breaches. Pathak et al. [Pathak et al(2021)] proposed a ZKP-based system to mitigate replay and MITM attacks, but its scalability is limited for large networks. Jamile et al. [Shahrouz and Analoui(2024)] combined ZKPs with blockchain for Vehicular Ad Hoc Networks (VANETs), ensuring anonymity and security, but with high computational overhead. Su et al. [Su et al(2024)] applied ZKPs to IoT authentication, offering efficiency for constrained devices, but limited to IoT applications.

Jiang et al. [Jiang and Guo(2024)] proposed an anonymous authentication scheme for IoV using blockchain, enhancing scalability and privacy, though its complexity hinders real-time performance. Grzonkowski et al. [Grzonkowski et al(2008)] focused on lightweight web authentication with ZKPs, minimizing overhead, but is vulnerable to advanced threats like phishing. Chistousov et al. [Chistousov et al(2022)] introduced an adaptive ZKP protocol that adjusts based on threat levels, improving security but adding latency.

The proposed ZKPVA Protocol builds on this research by addressing web-specific risks like identity theft and data leaks. Its modular design ensures strong security and efficiency, making it ideal for practical use.

3 Proposed Methodology

The methodology behind the proposed ZKPVA (Zero-Knowledge Proof-based Verification and Authentication) protocol is designed to provide privacy-preserving and secure authentication. The authentication process consists of several steps where the client and server exchange data to verify the client's identity without transmitting sensitive information, such as passwords, over the network. This exchange involves a challenge (a number generated by the server) and a response (a number calculated by the client for each challenge). The following section outlines the key steps in the authentication process, including the mathematical principles involved, and highlights the use of nonce and commitment schemes to enhance security. As shown in Fig. 1, the protocol is structured into four phases: Registration Phase, Commitment Phase, Challenge-Response Phase, and Verification Phase.

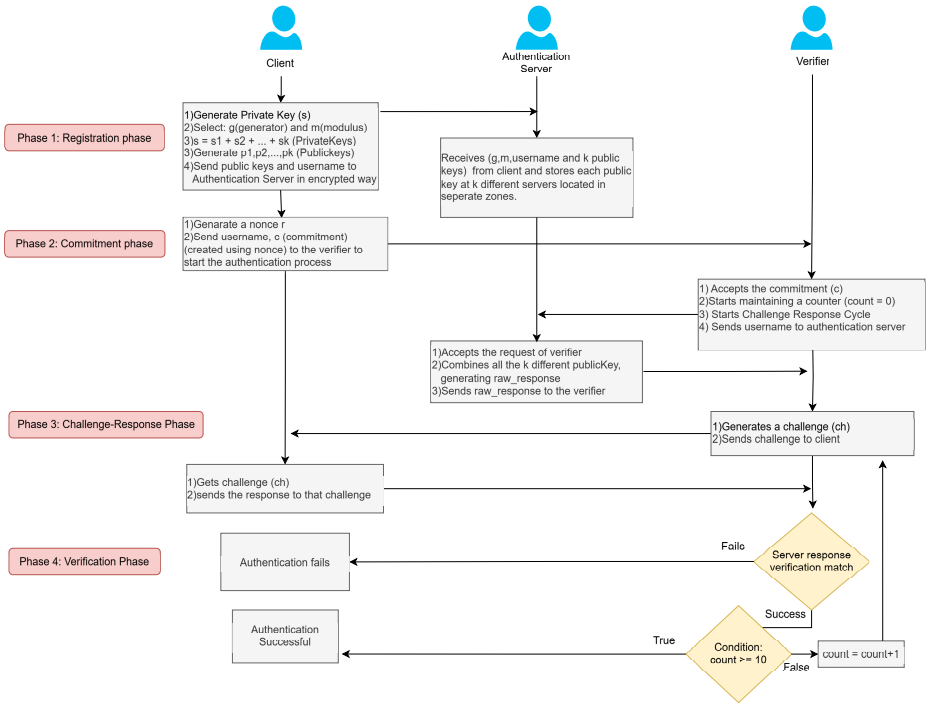

Fig. 1. ZKPVA Protocol

3.1 Registration Phase

As the first step, the client needs to register with the server and once share its information with the server through an encrypted or trusted channel. This information includes username, generator (g), modulus (m) and k public keys (p1,p2,...pk). The process starts by randomly selecting a private key (s), generator (g) (typically a small prime number), and modulus (m) (typically a large

prime number (2048 bits)). The private key s is used to create k secret keys (s1,s2,...sk) such that their sum is equal to s (s1+s2+...sk = s). These secret keys are used to calculate corresponding k public keys (p1,p2,...,pk).

$$p_i = g^{s_i} \mod m$$

where g is the generator, m is the modulus, and s_i is the i^{th} secret key and p_i is the i^{th} public key. The server stores this information for each client. The private key is kept secure by the client, while the information is made available to the server over network. server distributes this information to k different servers (data centers) located in different zones, each public key to one server(data center). This storage mechanism addresses the problem of data breaches.

3.2 Commitment Phase

At the start of the authentication process, client generates a secret random number (called nonce). The commitment ($c = (g^{nonce}) \mod m$) is a number that is sent to the verifier as a part of the authentication request along with username. The commitment is calculated using three numbers: secret random value (nonce) that was recently generated, The generator (g) (same as that used in registration phase) and The modulus (m) (also same as that used in the registration phase). The verifier than requests to the server to generate the raw-response which it generates as the product of the k public keys and sends to the verifier.

$$\text{raw-response rr} = p1 \cdot p2 \cdot ... \cdot pk \mod m$$

3.3 Challenge-Response Phase

A challenge (ch) is a number selected randomly by the server for the client that is smaller than modulus (m). A response (r) is a number that is calculated by the client as a response to the challenge received from the server.

Once the server receives the raw response from the authentication server, it generates a challenge (ch) which is sent to the client. The client is then expected to respond to this challenge by providing a response that proves they know the correct password without revealing it. Upon receiving the challenge, the client calculates a response using their secret private key (s) and salt.

$$\text{response r} = (g^{(nonce+(s*ch))}) \mod m$$

This response is then sent back to the verifier. The server verifies the response by comparing it to the expected response.

3.4 Verification Phase

To further enhance security, the proposed protocol involves multiple iterations of the challenge-response process, where both the client and verifier repeatedly

Algorithm 1 Zero-Knowledge Proof-based Authentication Protocol

Input: Generator g, Modulus m, Private key s, Secret keys $s_1, \ldots, s_k$, Public keys $p_1, \ldots, p_k$
Output: Authentication success or failure
procedure REGISTER(g, m, s)
    for $i = 1$ to k **do**
        Compute $p_i = g^{s_i} \mod m$
    end for
    Store and distribute $p_1, \ldots, p_k$
end procedure
procedure COMMIT($g, m,$ nonce)
    Client generates nonce and computes $c = g^{\text{nonce}} \mod m$
    Send c to verifier, who computes rr $= \prod_{i=1}^{k} p_i \mod m$
end procedure
procedure CHALLENGERESPONSE($g, m,$ nonce, ch)
    Client computes $r = g^{(\text{nonce}+s\cdot\text{ch})} \mod m$ and sends r
end procedure
procedure VERIFY($r,$ rr, ch, c, m)
    Compute expected-response er $= (\text{rr}^{\text{ch}} \cdot c) \mod m$
    if er == r **then**
        Authentication successful
    else
        Authentication failed
    end if
    Repeat for n iterations, authenticate if all pass
end procedure

exchange challenges and responses. This iterative process helps ensure that an attacker cannot impersonate the client by luck. the expected response for verification is calculated by the verifier

$$\text{expected-response er} = (rr^{ch} * c) \mod m,$$

here, rr is raw-response, ch is the challenge and c is commitment. Each time the server receives a valid response (er == r is True), it generates a new challenge and sends it to the client, repeating this step a predefined number of times (e.g., 10 iterations). After a successful set of iterations, the client is authenticated, and the server sends a confirmation message to the client along with any additional data associated with the user (e.g., account information).

4 Result Analysis and Discussion

4.1 Cryptanalysis of the ZKPVA Protocol

The proposed protocol is built on the foundation of the Discrete Logarithm Problem (DLP), which is computationally infeasible to solve using current classical

computing methods. By employing a large modulus (m) of at least 2048 bits and a generator (g), the protocol ensures that brute force or dictionary attacks are virtually impractical. Threshold cryptography, which splits the private key (s) into shares ($s_1, s_2, \ldots, s_k$), adds an additional layer of security by decentralizing sensitive information. Even if an attacker compromises fewer than k servers, the private key cannot be reconstructed, maintaining system integrity. The cryptographic commitment $c = g^{\text{nonce}} \mod m$ secures the random nonce and prevents sensitive information leakage during authentication. The modular arithmetic used in responses also prevents reverse engineering, solidifying the algorithm against cryptanalytic attacks.

4.2 Vulnerabilities Mitigation

The protocol addresses several vulnerabilities inherent in traditional authentication mechanisms. Replay attacks are mitigated through the use of a unique nonce for each session, ensuring that previously intercepted data cannot be reused. Distributing the private key into shares across multiple servers removes the risk of a single point of failure, which is critical in scenarios where server breaches are a concern. The Zero-Knowledge Proof mechanism ensures sensitive data such as passwords or private keys are never transmitted over the network, reducing the risk of interception. Additionally, the iterative challenge-response mechanism enhances security by ensuring that even a single compromised interaction does not jeopardize the overall process.

4.3 Attacks Mitigation

The implementation effectively mitigates multiple attack vectors:

- **Brute Force Attacks:** The large size of the modulus and the randomness introduced by the nonce and challenge ensure computational infeasibility for exhaustive search methods.
- **Man-in-the-Middle (MITM) Attacks:** The verifier cannot compute the response (r) without knowledge of both the private key (s) and the random nonce. The iterative nature of the protocol ensures that even partial interception does not compromise authentication.
- **Collusion Attacks:** Threshold cryptography ensures that collusion between fewer than k compromised servers cannot reconstruct the private key.
- **Impersonation Attacks:** The challenge-response mechanism ensures that only the legitimate client with the correct private key can compute valid responses.
- **Side-Channel Attacks:** The use of random nonces and commitment schemes reduces susceptibility to timing and power analysis attacks.

4.4 Real-Life Applications of the Proposed Protocol

The proposed protocol is versatile, applicable to various secure authentication scenarios:

- **Passwordless Login for Websites and Apps:** Users can prove that they have the password without ever sending it and can also help for secure and contactless online banking and other services online.
- **Automatic Authentication in IoT Devices:** Provides lightweight and secure authentication for resource-constrained IoT devices, using threshold cryptography and challenge-response mechanisms to prevent impersonation.
- **Secure access in High Risk Environments:** Employees accessing government services such as defense departments or corporate services or cloud services must log in securely, as stolen credentials can lead to data breaches. The protocol secures the system and allows access through decentralized key management, preventing single points of failure.

4.5 Performance Evaluation

The computation time of the entire authentication process depends mainly on the number of challenges in the challenge-response-verification cycle. As computations are kept efficient by the use of exponential operations and modular arithmetic. Also the time it takes to generate raw-response is also similar to the time it takes for a single server to respond because all the server who has part of the public key stored, send data simultaneously. Thus decreasing the number of challenges will result in lower response time but will lower the security of the protocol. Similarly with high number of challenges, the protocol can be made even more secure but will result in higher response time as well.

4.6 Security Analysis of the ZKPVA Protocol

The mathematical structure of the protocol ensures correctness and integrity:

- **Commitment Phase:** The client computes $c = g^{\text{nonce}} \mod m$, proving knowledge of the nonce without revealing it.
- **Challenge-Response Phase:** Upon receiving the challenge (ch), the client calculates the response as $r = g^{(\text{nonce}+s \cdot ch)} \mod m$, which is sent to the verifier.
- **Verification Phase:** The verifier computes the expected response $er = (rr^{ch} \cdot c) \mod m$, where rr is the raw response computed as the product of the public keys ($rr = p_1 \cdot p_2 \cdot ...p_k \mod m$) and (commitment $c = g^{nonce} \mod m$). Also ith public key is ($p_i = g^{s_i} \mod m$) and sum of k private keys ($s_1 + s_2 + ...s_k = s$). Thus the proof

$$er = (p_1 \cdot p_2 \cdot ... \cdot p_k)^{ch} \cdot g^{nonce} \mod m$$
$$er = (g^{s_1} \cdot g^{s_2} \cdot ... \cdot g^{s_k})^{ch} \cdot g^{nonce} \mod m$$
$$er = g^{(s_1+s_2+...+s_k) \cdot ch} \cdot g^{nonce} \mod m$$
$$er = g^{s \cdot ch + nonce} \mod m = \text{ r (response)}$$

If er matches the client's response r, authentication is successful.

– **Threshold Cryptography:** The private key s is split into shares $(s_1, s_2, \ldots, s_k)$, with each share used to compute a public key $p_i = g^{s_i} \mod m$. This ensures no single server holds the complete private key, and the complete response can only be reconstructed using all shares.

The logical linkage of phases ensures seamless integration: the commitment ensures integrity of the nonce, the challenge-response phase verifies the client's authenticity without revealing sensitive information, and the verification phase establishes trust through mathematical correctness. This cohesive structure forms a secure and efficient authentication protocol.

5 Conclusion and Future Scope of Enhancement

This paper introduces a novel Zero-Knowledge Proof (ZKP)-based verification and authentication (ZKPVA) protocol, providing a secure and efficient solution for identity verification across various domains. This protocol is also at the peak of convenience from the provers point of view who don't even need to click on a button for authentication. By leveraging ZKP's mathematical foundation, the protocol ensures privacy-preserving authentication without transmitting sensitive information like private keys or passwords, thus mitigating risks such as brute force, replay, impersonation, and collusion attacks.

Key features include threshold cryptography, which distributes private key components across multiple servers to minimize single points of failure, and an iterative challenge-response mechanism that prevents attackers from compromising the authentication process. Our design's scalability and efficiency make it suitable for both resource-constrained environments, such as IoT devices, and high-security applications like financial transactions and cloud computing.

Through detailed cryptographic analysis, we demonstrate that our approach effectively balances security, privacy and performance. The protocol's modularity allows adaptability to evolving cryptographic standards and emerging threats, ensuring it remains a future-proof solution.

While our ZKP-based protocol advances security and efficiency, future work could optimize or handle the high resource requirements with this protocol during raw-response generation, also where and how to store the private key securely as the protocol fails if the private key is compromised. Additionally, supporting multi-factor authentication and conducting real-world testing in various application domains will provide practical insights and will help refine the protocol for broader deployment, with industry collaboration essential for addressing domain-specific needs.

References

[Alrawili et al(2024)] Alrawili R, et al (2024) Comprehensive survey: Biometric user authentication application, evaluation, and discussion. Computers and Electrical Engineering 119:109,485

[Anyanwu et al(2024)] Anyanwu A, et al (2024) Data confidentiality and integrity: a review of accounting and cybersecurity controls in superannuation organizations. Computer Science & IT Research Journal 5(1):237–253

[Birthriya et al(2025)] Birthriya SK, et al (2025) Detection and prevention of spear phishing attacks: A comprehensive survey. Computers & Security p 104317

[Chistousov et al(2022)] Chistousov NK, et al (2022) Adaptive authentication protocol based on zero-knowledge proof. Algorithms 15(2)

[Choudhary et al(2024)] Choudhary S, et al (2024) Pkif-aka: a public key infrastructure free authenticated key agreement protocol for smart grid communication. IETE Journal of Research 70(4):3395–3406

[Grzonkowski et al(2008)] Grzonkowski S, et al (2008) Extending web applications with a lightweight zero knowledge proof authentication. In: Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, p 65–70

[Hoffman(2024)] Hoffman A (2024) Web application security. " O'Reilly Media, Inc."

[Jiang et al(2024)] Jiang C, et al (2024) An efficient privacy-preserving scheme for weak password collection in internet of things against perpetual leakage. IEEE Transactions on Information Forensics and Security

[Jiang and Guo(2024)] Jiang W, Guo Z (2024) An anonymous authentication scheme for internet of vehicles based on trug-pbft master-slave chains and zero-knowledge proof. IEEE Internet of Things Journal pp 1–1

[Kantipudi et al(2024)] Kantipudi R, et al (2024) A comprehensive analysis on using multifactor authentication system for three level security. In: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), IEEE, pp 498–503

[Muzammil et al(2024)] Muzammil MB, et al (2024) Unveiling vulnerabilities of web attacks considering man in the middle attack and session hijacking. IEEE Access

[Pathak et al(2021)] Pathak A, et al (2021) Secure authentication using zero knowledge proof. In: 2021 Asian Conference on Innovation in Technology (ASIANCON), pp 1–8

[Sahoo et al(2024)] Sahoo AK, et al (2024) Vulnerability analysis of low-entropy face recognition systems against brute force attacks. In: 2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), IEEE, pp 1–9

[Shahrouz and Analoui(2024)] Shahrouz JK, Analoui M (2024) An anonymous authentication scheme with conditional privacy-preserving for vehicular ad hoc networks based on zero-knowledge proof and blockchain. Ad Hoc Networks 154:103,349

[Su et al(2024)] Su Z, et al (2024) A secure and efficient authentication scheme for large-scale iot devices based on zero-knowledge proof. Electronics 13(18)

[Veeramachaneni(2025)] Veeramachaneni V (2025) Emerging authentication technologies for zero trust in iot systems. Journal of Advance Research in Mobile Computing 7(1):7–21

[Wu et al(1998)] Wu TD, et al (1998) The secure remote password protocol. In: NDSS, Citeseer, vol 98, p 111

[Zhou et al(2024)] Zhou L, et al (2024) Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges and opportunities. Journal of Information Security and Applications 80:103,678

[Zhuravel(2024)] Zhuravel D (2024) Enhancing website security: A comparative study of oauth, saml, and their integration into helpme office hours system