2024 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC) | 979-8-3503-6900-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/MoNeTec60984.2024.10768120

An Anonymous Authentication Scheme Based on Blind Signatures for the FIDO Protocol

Fan Dang
Global Innovation Exchange
Tsinghua University
Beijing, China
dangfan@tsinghua.edu.cn

Xikai Sun
Department of Automation
Tsinghua University
Beijing, China
sxk23@mails.tsinghua.edu.cn

Kebin Liu
Global Innovation Exchange
Tsinghua University
Beijing, China
kebinliu2021@tsinghua.edu.cn

Xuan Ding
School of Software
Tsinghua University
Beijing, China
dingxuan@tsinghua.edu.cn

Xu Wang
Global Innovation Exchange
Tsinghua University
Beijing, China
xu_wang@tsinghua.edu.cn

Yunhao Liu
Department of Automation
Tsinghua University
Beijing, China
yunhao@tsinghua.edu.cn

***Abstract*—The Fast Identity Online (FIDO) authentication protocol, the latest iteration of the FIDO2 standard, aims to provide a more secure and user-friendly online authentication method. In the era of digital transformation, anonymity has become a critical aspect of digital security. This paper proposes a novel anonymous authentication extension for FIDO based on blind signatures, enabling users to obtain signatures on public keys without revealing their content to the relying party. The proposed scheme is evaluated in detail, demonstrating its feasibility and effectiveness. By addressing a significant gap in current authentication methods and enhancing user privacy, this research contributes to the advancement of secure and anonymous online authentication techniques.**

***Index Terms*—FIDO2, anonymous authentication, blind signature**

I. Introduction

FIDO2 represents the latest advancement in the Fast Identity Online (FIDO) authentication standard, with the goal of providing a more secure and user-friendly method for authenticating users online. Unlike traditional password-based approaches, FIDO2 utilizes public-key cryptography, allowing users to authenticate without divulging sensitive information to the server. This standard has garnered widespread adoption by major technology companies and is seen as a significant step towards a password-less future [1].

FIDO2 has been implemented on various devices, showcasing its popularity and promising solutions. For example, YubiKey, a hardware security key, provides strong two-factor authentication and supports FIDO2 protocols [2]. Similarly, CanoKey, an open source security key, offers FIDO2 compatibility and improves security for online accounts [3]. In addition, Passkey [4], a FIDO2 authenticator mixed with hardware / software, allows users to authenticate using their mobile devices, providing a convenient and secure alternative to traditional passwords.

Fig. 1. Examples of FIDO2 devices: YubiKey, CanoKey, and Passkey

In today's era of digital transformation, anonymity has emerged as a crucial component of digital security. It serves to protect users' identities, safeguarding them from potential threats such as identity theft and targeted attacks. Anonymity also upholds privacy, a fundamental human right, by empowering individuals to engage online without fear of surveillance or discrimination [5]. The importance of privacy computing has become increasingly evident in the digital age. With the proliferation of data and the growing reliance on online services, protecting users' personal information has become paramount. In scenarios necessitating anonymous authentication, three parties are typically involved: the user (or prover), the relying party, and a verifier. However, the standard FIDO2 protocol focuses on a two-party model involving solely the user and the relying party, without the inclusion of a third-party verifier.

Regrettably, integrating anonymity into FIDO2 presents several challenges. The standard FIDO2 design prioritizes identity verification between the user and the relying party, neglecting the inclusion of a third-party verifier. This two-party model

conflicts with the objective of anonymous authentication, which often requires a three-party interaction. The specific challenges include:

- **Compatibility with Existing Systems**: Adapting FIDO2 to include a verifier without disrupting existing user-relying party interaction is complex.
- **Maintaining Anonymity**: Ensuring that the user's identity remains concealed from the verifier while still allowing for valid authentication requires careful design.
- **Balancing Security and Anonymity**: Achieving a delicate balance between robust security and user anonymity without compromising either aspect is a significant hurdle.

A potential solution to these challenges lies in the use of blind signatures. A blind signature is a cryptographic technique that allows a message to be signed without revealing its content to the signer. This property makes it an ideal tool for ensuring anonymity in various applications, including authentication processes. By integrating blind signatures with FIDO2, it becomes feasible to authenticate users without compromising their anonymity, aligning with the increasing demand for privacy-conscious solutions.

The proposed scheme in this paper aims to extend FIDO2 with an anonymous authentication extension based on blind signatures. The process is summarized as follows:

1) Each user generates a public-private key pair and blinds the public key p to obtain p', which is sent to the relying party via the verifier. The relying party checks the user's identity using FIDO2 authentication and then signs the blind public key p'.
2) The user unblinds the received signature on the blinded public key, obtaining the relying party's signature on the public key.
3) The user also signs the verifier's provided challenge using the private key from Step 1. The verifier can verify the signature, ensuring that it is from the user and is in line with the FIDO2 protocol.

This research makes a significant contribution to the field by addressing a notable gap in current authentication methods. It offers a comprehensive analysis of the challenges and solutions related to integrating anonymity into FIDO2 and presents a novel scheme that leverages blind signatures. The paper also includes a detailed evaluation of the proposed extension, demonstrating its feasibility and effectiveness. By advancing the understanding of anonymous authentication and providing a tangible solution, this work represents a meaningful step towards a more secure and privacy-respecting digital world.

In the remainder of the paper, we first comprehensively review the previous research in Section II, followed by a detailed description of the theoretical framework of the design in Section III. The extension design of how we integrate blind signatures into FIDO2 is elaborated in Section IV. In Section V, we discuss the open-source platform we used to implement the extension. In Section VI, we provide a thorough demonstration and discussion of a series of experimental results with the implementation, and the conclusions are finally drawn in Section VII.

II. Related Work

The quest for secure and anonymous authentication has been a topic of interest for researchers for several years. This section reviews the previous works in the areas of authentication methods in FIDO2, blind signatures in cryptography, anonymous authentication, and identifies the gaps in the current research.

A. Authentication Methods in FIDO2

The FIDO2 protocol has emerged as a significant advancement in the realm of user authentication. It aims to provide stronger, simpler public key-based credentials for online services, eliminating the need for passwords. Lyastani et al. [6] explored the user-friendliness of FIDO2 passwordless authentication, suggesting its potential to replace traditional methods1. Kunke et tal. [7] evaluated the strategies for account recovery with FIDO2-based passwordless authentication, emphasizing the importance of user experience in the recovery process. The concept of continuous web authentication was introduced by Klieme et al. [8], which proposed an extension to the FIDO2/WebAuthn protocol to support continuous user authentication. The distinction between platform and roaming authentication on smartphones was also explored, highlighting the advantages and challenges of each approach [9].

B. Blind Signatures in Cryptography

Blind signatures play a pivotal role in cryptography, ensuring the signer remains unaware of the content they are signing. The foundational work *blind signatures for untraceable payments* introduced the concept, emphasizing its application in untraceable payment systems [10]. Over the years, various blind signature schemes have been proposed. For instance, a scheme based on the discrete logarithm problem was introduced, offering robust security features [11]. Another notable development is the lattice-based blind signatures, which leverage the hardness of lattice problems to ensure security [12]. These signatures have also found applications in secure e-voting systems, ensuring voter privacy while maintaining the integrity of the vote [13].

C. Anonymous Authentication

Anonymous authentication ensures user privacy during the authentication process. The concept of *k-times anonymous authentication* was introduced, allowing users to authenticate themselves anonymously for a predefined number of times [14]. Camenisch et al. [15] proposed an efficient periodic n-times anonymous authentication mechanism. The application of anonymous authentication in various domains has also been explored. For instance, a robust anonymous authentication protocol tailored for health-care applications using

wireless medical sensor networks was proposed, emphasizing the importance of patient data privacy [16]. Furthermore, the decentralized access control with anonymous authentication for data stored in clouds was introduced, highlighting the significance of user privacy in cloud environments [17]. Anonymous authentication can also improve the security level of contactless payments [18], [19].

D. Gaps in Current Research

While significant advancements have been made in FIDO2 authentication and anonymous authentication, there remain gaps in the current research. Kepkowski et al. [20] highlighted the usability challenges faced by enterprises when implementing FIDO2. Hanzlik et al. [21] formalized the privacy and revocation aspects for FIDO2, suggesting areas of improvement. The misconceptions users have about FIDO2 Biometric WebAuthn were explored in a study, emphasizing the need for better user education and awareness [22]. These gaps present opportunities for further research and development in the domain of FIDO2 and anonymous authentication.

III. Theoretical Framework

A. Basics of FIDO2 Protocol

The FIDO2 protocol is a cutting-edge open standard for authentication that aims to revolutionize the way users authenticate themselves online. It offers a more secure and user-friendly alternative to traditional password-based authentication methods, which have become increasingly vulnerable to various security threats such as phishing, password reuse, and data breaches. FIDO2 consists of two main components: the Web Authentication API (WebAuthn) and the Client to Authenticator Protocol 2 (CTAP2).

WebAuthn is a powerful API that enables online services to leverage registered devices as authenticators. This means that users can authenticate themselves using their personal devices, such as smartphones, laptops, or security keys, without the need to remember complex passwords. WebAuthn is supported by major web browsers, making it widely accessible to users and developers alike.

On the other hand, CTAP2 is a protocol that facilitates the interaction between external devices, such as security keys, and web browsers. This allows users to authenticate themselves using a physical device that they possess, providing an additional layer of security. CTAP2 is designed to be simple and intuitive, making it easy for users to adopt and use.

The FIDO2 protocol operates in a challenge-response manner, ensuring that the authentication process is secure and tamper-proof. When a user attempts to authenticate, the relying party (i.e., the online service) sends a challenge to the user's authenticator (e.g., a security key or a smartphone). The authenticator then signs the challenge using its private key, which is securely stored within the device. The signed challenge, along with the authenticator's public key, is sent back to the relying party for verification. The relying party uses the public key to verify the signature, ensuring that the user is in possession of the corresponding private key, thus authenticating their identity.

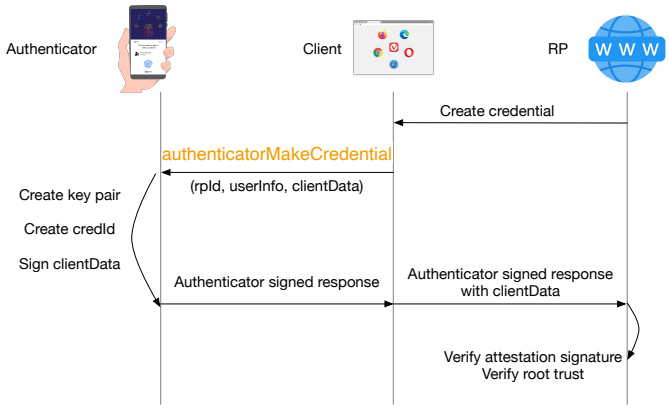

Fig. 2. Registration Flow of FIDO2 (authenticatorMakeCredential)

Two key operations within the FIDO2 protocol are:

1) The `authenticatorMakeCredential` operation, which is used for registering a new authenticator with an online service. During this process, the authenticator generates a new public-private key pair specifically for that service. The online service sends a challenge to the authenticator, which signs the challenge using its newly generated private key. The signed challenge, along with the public key and other relevant information, is sent back to the online service. This allows the service to create a unique credential for the authenticator, securely establishing its identity for future authentications. The registration flow is illustrated in Fig. 2.
2) The `authenticatorGetAssertion` operation, which is used for user authentication. When a user attempts to log in to an online service, the service sends a challenge to the user's authenticator. The user then provides their authentication factor, such as biometric data or a PIN, to unlock the authenticator. Upon successful verification of the user's authentication factor, the authenticator signs the challenge using its private key associated with the online service. The signed challenge is sent back to the online service, which then verifies the signature using the previously stored public key. If the verification is successful, the user is granted access to the service. This process ensures secure and convenient user authentication, as it relies on the possession of a physical device and the user's unique authentication factor. The authentication flow is depicted in Fig. 3.

The adoption of FIDO2 has been growing rapidly, with major technology companies such as Google, Microsoft, and Apple integrating it into their products and services. As more online services and users embrace FIDO2, it has the potential

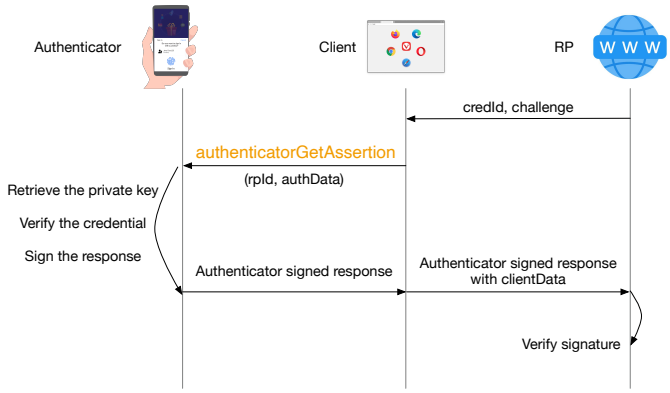

Fig. 3. Authentication Flow of FIDO2 (authenticatorGetAssertion)

to become the new standard for online authentication, providing a more secure and user-friendly alternative to passwords.

B. Principles of Blind Signatures

Blind signatures are a cryptographic technique that allows a person to obtain a signature on a document without revealing the content of the document to the signer. This concept, introduced by David Chaum [10], plays a crucial role in ensuring privacy and anonymity in various applications, such as electronic voting, digital cash systems, and anonymous communication.

The primary goal of blind signatures is to enable a user to request a signature on a message from a signer while keeping the message content hidden from the signer. This is achieved through a clever use of mathematical operations that ”blind” the message before sending it to the signer.

In a typical blind signature scheme, the user first blinds the message using a blinding factor, which is a random value chosen by the user. The blinding process involves applying a mathematical function to the message and the blinding factor, resulting in a blinded message. The user then sends the blinded message to the signer, who signs it using their private key without knowing the actual content of the message. The signer returns the signed blinded message to the user.

Upon receiving the signed blinded message, the user unblinds it by removing the blinding factor. This unblinding process reveals a valid signature on the original message, which the user can then use for various purposes, such as proving the authenticity of the message or engaging in anonymous transactions.

Blind signatures provide two essential properties: unforgeability and unlinkability. Unforgeability ensures that only the legitimate signer can produce valid signatures, preventing anyone else from creating fake signatures. This property is crucial to maintaining the integrity and trustworthiness of the system. Unlinkability, on the other hand, ensures that the signer cannot link the blinded message they signed to the unblinded signature that the user later reveals. This property is essential for preserving the user’s privacy and anonymity, as the signer cannot trace the signature back to the original message or the user who requested it.

The security of blind signature schemes relies on the underlying mathematical assumptions and the proper implementation of blinding and unblinding operations. Various blind signature schemes have been proposed, each with its own strengths and weaknesses. Some notable examples include the RSA blind signature scheme, the Schnorr blind signature scheme, and the Chaum-Pedersen blind signature scheme.

Blind signatures have found numerous applications in privacy-preserving systems. In electronic voting systems, blind signatures can be used to ensure that voters can cast their votes anonymously while still allowing the authorities to verify the validity of the votes. In digital cash systems, blind signatures enable users to engage in untraceable financial transactions, protecting their financial privacy. Blind signatures are also used in anonymous credential systems, where users can prove their attributes or qualifications without revealing their identity.

C. Combining FIDO2 with Blind Signatures

The integration of FIDO2 with blind signatures presents a groundbreaking approach to anonymous authentication, offering a unique blend of security and privacy. By seamlessly incorporating blind signatures into the FIDO2 authentication process, users gain the ability to obtain signatures on their public keys without exposing them to the relying party. This innovative combination opens up new possibilities for secure and privacy-preserving authentication in various digital environments.

The fusion of FIDO2 and blind signatures elegantly maintains the robust security features that FIDO2 is renowned for while introducing an additional layer of privacy protection. The relying party can still confidently verify the user’s identity and the authenticity of the public key, ensuring the integrity of the authentication process. However, the critical difference lies in the fact that the relying party cannot establish a direct link between the public key and a specific user. This separation of identity and public key enhances user privacy and mitigates the potential for tracking or profiling based on authentication data.

In this combined approach, the user initiates the process by blinding their public key using advanced cryptographic techniques. The blinded public key is then securely transmitted to the relying party, following the standard FIDO2 authentication flow. Upon receiving the blinded public key, the relying party performs the necessary verification steps and signs the blinded key, unaware of its true contents. The user, in turn, unblinds the signature received from the relying party, ultimately obtaining a valid signature on their original public key. This intricate dance of blinding and unblinding ensures that the user’s anonymity remains intact while the integrity of the public key is preserved.

The integration of blind signatures into the FIDO2 framework aligns seamlessly with the core principles of both technologies. FIDO2 prioritizes strong, multi-factor authentication and aims to provide a secure and user-friendly alternative to traditional password-based systems. Blind signatures, on the other hand, focus on preserving user privacy by allowing for the verification of information without revealing its contents. By bringing these two paradigms together, the combined approach offers a powerful solution that addresses both security and privacy concerns in an increasingly digital world.

The potential applications of this combined approach are vast and far-reaching. From online transactions and identity verification to secure communication and anonymous credentials, the integration of FIDO2 with blind signatures can revolutionize the way we authenticate and interact in digital spaces. It provides a solid foundation for building trust and confidence in online interactions while empowering users to maintain control over their personal information.

IV. Extension for Anonymous Authentication

We now give the design of the anonymous authentication extension.

A. Entities

The anonymous authentication extension is designed to ensure that users can prove their identity without revealing their actual identity details. This is achieved through the collaboration of three main entities, each with its distinct role and responsibilities.

- **User (Authenticator):** At the core of this system is the user with an authenticator, who has an account in an online service. This account is secured by a strong authenticator, which ensures that only the legitimate user can access it. The user's primary goal is to prove their identity to various services without revealing specific details about themselves. This is especially crucial in scenarios where privacy and anonymity are of the utmost importance.
- **Relying Party:** The relying party is typically a service provider, such as a website or an online service. When a user wishes to prove their identity to a third party, the relying party facilitates this by interacting with the authenticator to create proof. However, it is essential that the relying party remains unaware of how the verifier validates the user. This ensures that while the relying party can attest to the user's authenticity, it does not have insights into the verification process, thereby maintaining the user's privacy.
- **Verifier:** The verifier plays a pivotal role in this system. It is responsible for initiating the authentication process and ensuring that a user is indeed a valid user by checking the signature provided during the authentication. The verifier receives the signed challenge and verifies the final signature of the challenge. However, a fundamental principle of this system is that while the verifier can confirm the authenticity of the user, it should not under any circumstances know the actual identity of the user. This ensures that the user's identity remains protected, and only their validity is confirmed.

In essence, this system is a delicate balance of trust and anonymity. The User trusts the RP with their account details, but seeks to remain anonymous when proving their identity to other services. The RP, while facilitating this, remains unaware of the verification process. The Verifier, on the other hand, ensures the user's authenticity but remains blind to their actual identity. This intricate dance of trust and privacy ensures that users can navigate the digital world securely while maintaining their anonymity.

B. Protocol Flow

The architecture of the anonymous authentication extension revolves around the intricate interactions between three primary entities: the verifier, the relying party, and the authenticator. These entities work in harmony to ensure a secure and efficient authentication process while preserving the anonymity of the user. The flow of operations among these entities, as depicted in Fig. 4, unfolds in a series of carefully orchestrated steps:

1) The relying party initiates the process by requesting the authenticator to create a credential. This request sets the gears in motion, prompting the authenticator to generate a key pair and a random value r specifically designed for anonymous authentication. These newly generated components are then securely associated with the credential, forming a robust foundation for the subsequent steps.
2) Upon receiving a challenge from the verifier, the authenticator springs into action. It deftly signs the challenge using its private key, demonstrating its authenticity and commitment to the process. Simultaneously, the authenticator computes a blind digest of the public key, a crucial step that will later ensure the anonymity of the user (detailed in Section IV-C). The resulting authentication data, a comprehensive package containing the signature of the authentication data (`sigAuth`), the signed challenge (`sigChal`), the public key (`pub`), and the blind digest of the public key (`digestPub`), is then securely transmitted back to the verifier.
3) The verifier, acting as a conduit, promptly forwards the received authentication data, `sigAuth`, and `digestPub` to the relying party. The relying party, being the trusted entity, meticulously verifies the signature of the authentication data, ensuring the integrity and authenticity of the information. Upon successful verification, the relying party generates a blind signature of the challenge, a cryptographic seal of approval, which is then handed back to the verifier.
4) In the final leg of the journey, the verifier reaches out to the authenticator, requesting the removal of the blinding

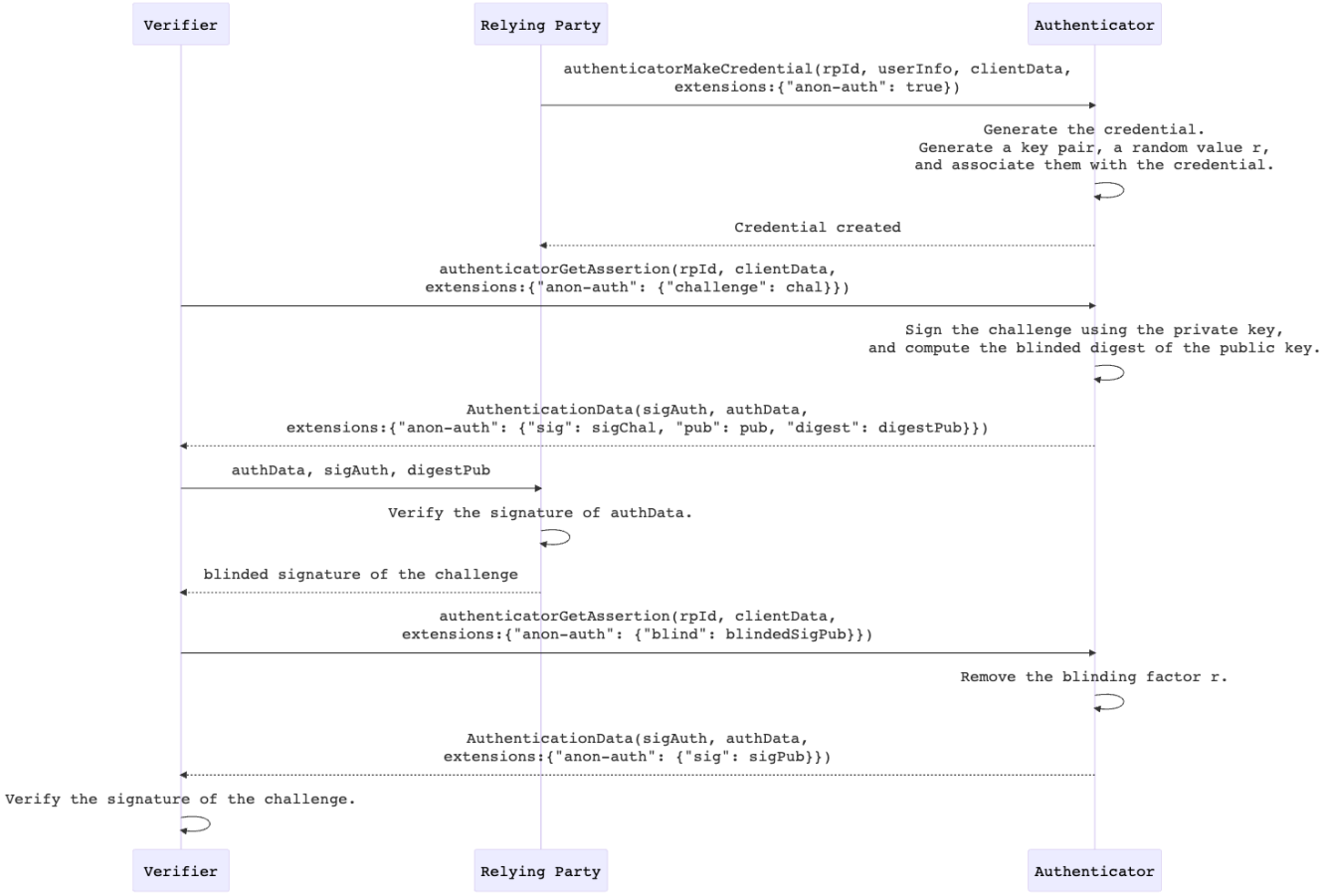

Fig. 4. Overview of the anonymous authentication workflow.

factor r from the blind signature of the public key. The authenticator, being the keeper of secrets, complies with the request and carefully removes the blinding factor, ensuring that the user's anonymity remains intact. The authenticator then sends back the updated authentication data, which now includes the unblind signature of the public key (`sigPub`). As a final measure of security, the verifier verifies the signature of the challenge, confirming the authenticity and integrity of the entire process.

This intricate dance of authentication, with each entity playing its role to perfection, ensures that the user's identity remains shielded while granting access to the desired resources. The anonymous authentication extension, through its carefully designed protocol flow, strikes a delicate balance between security and privacy, providing a seamless and trustworthy experience for all parties involved.

C. Algorithms

The extension employs the RSA cryptographic algorithm to achieve anonymous authentication, ensuring the privacy and security of the authenticator's identity. The RSA algorithm is a widely used public-key cryptosystem that provides robust security based on the difficulty of factoring large integers. The steps involved in the anonymous authentication process are as follows:

1) **Key Generation:** The authenticator utilizes the RSA algorithm to generate an RSA key pair, which consists of a public key and a private key. The relying party's public key is denoted as (n, e), where n is the modulus (the product of two large prime numbers) and e is the public exponent. The corresponding private key is denoted as (n, d), where d is the private exponent. The generation

of the key pair is a crucial step in ensuring the security of the authentication process.

2) **Digest Computation:** The authenticator computes the digest of the public key, denoted as x. The digest is typically obtained by applying a cryptographic hash function, such as SHA-256, to the public key. The hash function takes the public key as input and produces a fixed-size output, which serves as a unique representation of the key. The digest is used in the subsequent steps of the authentication process.
3) **Blinding the Digest:** To protect the privacy of the authenticator's public key, the authenticator blinds the digest x using a blinding factor. The blinding process is performed using the formula:

$$x' = x \times r^e \bmod n, \tag{1}$$

where r is a random value generated by the authenticator, and e is the public exponent of the relying party's public key. The blinding factor r is chosen randomly and kept secret by the authenticator. The modular exponentiation operation ($r^e \bmod n$) is performed to create the blinded digest x'.
4) **Signature by Relying Party:** The relying party receives the blinded digest x' from the authenticator and signs it using its private key. The signature is computed as:

$$y' = (x')^d \bmod n = (x^d \times r^{ed}) \bmod n = (x^d \times r) \bmod n. \tag{2}$$

The relying party uses its private exponent d to perform the modular exponentiation operation on the blinded digest x'. The resulting value y' is the signed blinded digest, which is sent back to the authenticator.
5) **Unblinding the Signature:** Upon receiving the signed blinded digest y' from the relying party, the authenticator unblinds it to obtain the actual signature. The unblinding process is performed using the formula:

$$y = y' \times r^{-1} \bmod n = x^d \bmod n. \tag{3}$$

The authenticator multiplies the signed blinded digest y' by the modular multiplicative inverse of the blinding factor r (denoted as r^{-1}) modulo n. This operation removes the blinding factor and reveals the actual signature y of the digest x. The resulting signature y can be verified using the relying party's public key.

The anonymous authentication process described above ensures that the authenticator can obtain a valid signature on its public key without revealing the actual key to the relying party. The blinding technique prevents the relying party from learning the authenticator's public key during the signing process. Furthermore, the use of the RSA algorithm provides strong security guarantees, as the private key remains protected and is not exposed during the authentication process.

The unblinded signature y can be verified by anyone using the relying party's public key. The verification process involves computing the digest of the authenticator's public key and comparing it with the decrypted signature using the relying party's public key. If the two values match, it confirms that the signature is valid and that the authenticator possesses the corresponding private key.

By employing these cryptographic techniques, the extension enables anonymous authentication, preserving the privacy of the authenticator while still allowing the relying party to verify the authenticity of the authenticator's credentials. This approach provides a secure and privacy-preserving mechanism for authentication in various applications and scenarios.

D. System Goals

The design of our anonymous authentication extension aims to address several pivotal goals to ensure both security and user privacy. These goals serve as guiding principles throughout the development and implementation phases:

1) **User Anonymity:** Foremost, the system prioritizes the preservation of user anonymity. While users should be able to prove their identity and access services, their specific identity details should remain concealed. This ensures that users can interact with online platforms without the risk of their personal information being exposed or misused.
2) **Robust Security:** Beyond anonymity, the system is designed to provide a robust layer of security. By leveraging advanced cryptographic techniques and secure communication protocols, the system ensures that user data, even in its anonymized form, is protected from potential threats and breaches.
3) **Interoperability:** Recognizing the diverse digital landscape, the system is designed for seamless integration with various online platforms. Whether it's a website, an online service, or a mobile application, the system's design ensures compatibility and smooth operation.
4) **Efficiency:** Speed and responsiveness are crucial for user experience. The system aims to provide swift authentication processes without compromising on security. By optimizing cryptographic operations and streamlining communication between entities, users can expect a rapid yet secure authentication experience.
5) **Transparency:** While the system operates behind the scenes, it is essential that users and service providers understand its workings. The design emphasizes transparency, ensuring that all parties involved are aware of the authentication process's flow and the measures in place to protect user anonymity and data.

E. Security Analysis

1) Anonymity: The proposed protocol leverages the concept of blinding to ensure the anonymity of the authenticator's public key during the signing process. By employing a blinding technique, the relying party is presented with a blinded digest x' instead of the original digest x. This crucial step prevents

the relying party from gaining knowledge of the authenticator's public key, thereby preserving its anonymity. The relying party proceeds to sign the blinded digest x' without any awareness of the original digest x, further reinforcing the anonymity of the authenticator's public key.

Moreover, the protocol maintains a separation of knowledge between the relying party and the verifier. While the relying party attests to the authenticity of the user, the verifier can only validate the user's legitimacy through the relying party's confirmation. The FIDO2 public key, which is essential for establishing the user's identity, remains exclusively known to the relying party and is not exposed to the verifier. This separation ensures that the verifier cannot discern the user's specific identity, even though they can confirm the user's signature on the challenge.

It is important to note that the relying party, despite its involvement in the authentication process, cannot forge the user's signature. The blinding mechanism prevents the relying party from manipulating the signature, as they do not possess the necessary information to generate a valid signature on behalf of the user. Furthermore, the verifier remains oblivious to the user's identity throughout the process, as they only receive the attestation from the relying party without any direct exposure to the user's public key.

2) Integrity: The integrity of the authentication process is safeguarded through the utilization of the RSA digital signature mechanism. When the relying party signs the blinded digest x', they are effectively attesting to the authenticity of the original digest x. This signature serves as a tamper-evident seal, ensuring that the digest x has not been modified or altered during the process.

Upon receiving the signed blinded digest, the authenticator proceeds to unblind the signature, resulting in y. The unblinded signature y can then be validated by the verifier using the relying party's public key (n, e). The successful validation of the signature provides assurance that the signature was indeed generated by the relying party and that the digest x has remained intact throughout the process.

The RSA digital signature mechanism employed in this protocol is known for its robustness and security properties. The mathematical foundation of RSA ensures that it is computationally infeasible for an attacker to forge a valid signature without knowledge of the relying party's private key. Furthermore, any tampering or modification of the digest x would invalidate the signature, making it detectable during the verification process. Thus, the integrity of the authentication process is maintained, providing confidence in the authenticity of the user and the validity of the signed challenge.

3) Confidentiality: The protocol ensures the confidentiality of sensitive information, such as the authenticator's public key and the associated digest, through the use of blinding and unblinding techniques. The blinding process transforms the original digest x into a blinded digest x' using a random value r known only to the authenticator. This blinded digest x' does not reveal any information about the original digest x or the public key associated with it.

The confidentiality of the blinding factor r is crucial to the security of the protocol. By keeping r secret and known only to the authenticator, it becomes infeasible for any unauthorized party to unblind the signed digest. Even if an attacker were to intercept the blinded digest x' during transmission, they would not be able to recover the original digest x without knowing of the blinding factor r.

Furthermore, the unblinding process, which is performed by the authenticator after receiving the signed blinded digest from the relying party, ensures that the relying party cannot learn any information about the original digest x or the authenticator's public key. The unblinding operation effectively removes the blinding factor r from the signed blinded digest, resulting in a valid signature y in the original digest x. This unblinded signature can be safely shared with the verifier without compromising the confidentiality of the authenticator's public key.

The combination of blinding and unblinding techniques employed in this protocol guarantees that the sensitive information remains confidential throughout the authentication process. The relying party, verifier, and any potential adversaries are prevented from gaining unauthorized access to the authenticator's public key or the original digest, thereby maintaining the confidentiality of the user's identity and the integrity of the authentication process.

V. Implementation

We have implemented the communication between the relying party and the verifier using Golang and gRPC.

The core of our FIDO2 implementation is built upon CanoKey [23], an open-source security key platform. CanoKey offers a comprehensive and tailored environment for FIDO2 operations, providing a secure and efficient foundation for the authentication process. By integrating CanoKey into our implementation, we harness its robust features and ensure that the core functionalities of FIDO2 are seamlessly incorporated while adding the anonymous authentication extension. CanoKey's well-designed architecture and strong security measures contribute to the overall reliability and integrity of our implementation.

One significant challenge encountered during the implementation process is the lack of native support for customized extensions in mainstream browsers. To overcome this obstacle, we have utilized a modified version of the libfido2 [24] library. This adapted library is specifically designed to accommodate and support the *anon-auth* extension, enabling seamless integration of anonymous authentication capabilities. By leveraging this modified library, we ensure that critical processes such as blinding, signing, and unblinding are executed correctly and securely during the authentication flow. The integration of the modified libfido2 library allows us to bridge the gap between

our custom extension and the existing browser infrastructure, providing a smooth and secure user experience.

To optimize the performance of cryptographic operations, particularly those involving RSA 4096, we have incorporated the xRSA algorithm [25] into our implementation. To address this challenge, the xRSA algorithm is strategically designed to build upon an RSA 2048 accelerator. By leveraging the efficiency of RSA 2048 operations, xRSA enables us to perform RSA 4096 computations more effectively. This approach not only optimizes the overall performance of RSA operations but also ensures that critical processes such as key generation, blinding, signing, and unblinding are executed securely and efficiently. The integration of the xRSA algorithm enhances the scalability and practicality of our implementation, making it suitable for real-world deployments.

VI. Evaluation

In this section, we evaluate the cost of the *anon-auth* extension to end users and service providers, focusing on the computational overhead and its impact on the overall performance of the authentication process.

Experiment setup. To ensure a controlled and consistent environment, allowing for accurate and reliable results, we leverage Amazon AWS EC2 instances for our experiments. The verifier and the relying party are deployed on a `c5.4xlarge` instance, which is equipped with 8 cores, each having 2 hyperthreads, totaling 16 threads. The instance also possesses 32GB of memory, providing ample resources for the evaluation. To simulate a realistic network condition, we configure the connection between the verifier and the relying party to have a bandwidth of 1000 Mbps, which is representative of a high-speed network environment.

Computational Overhead. One of the primary concerns when introducing a new authentication mechanism is the computational overhead it imposes on the system. The proposed *anon-auth* design integrates blind signatures into the FIDO2 protocol, which inherently adds cryptographic operations to the authentication process. These additional operations have the potential to impact the overall performance and user experience. To assess the feasibility and practicality of the proposed scheme, it is crucial to evaluate the computational overhead incurred by the additional cryptographic operations and compare it with the standard FIDO2 protocol.

A. Enrollment

In this scheme, the generation of the public-private key pair plays a crucial role. However, a typical FIDO2 credential utilizes an elliptic-curve cryptographic (ECC) algorithm, which operates much faster than the RSA key scheme. Hence, the time required for this operation could be a significant factor. The result is displayed in Fig. 5. The enrollment process consists of two steps: the FIDO2 credential generation and the blinded key pair generation. The average running times for these steps are 247 ms and 2027 ms, respectively. Fortunately, the enrollment only needs to be executed once, making the cost acceptable. It is important to note that the enrollment process is a one-time setup and does not impact the real-time authentication performance. Moreover, the enrollment can be performed offline, allowing users to complete the process at their convenience without relying on a continuous network connection.

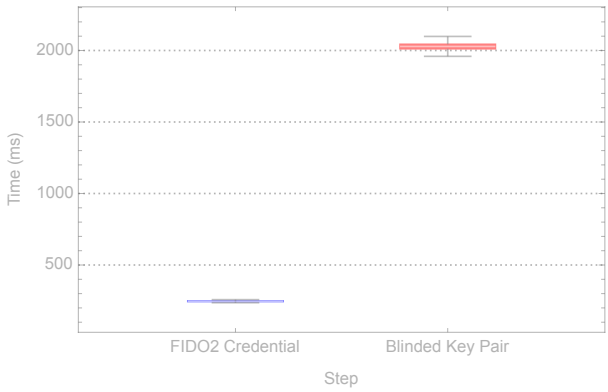

Fig. 5. The average time taken for enrollment, including two steps: credential generation and blinded key pair generation

B. Blinding and Unblinding Operations

Blinding the public key and subsequently unblinding the received signature are crucial steps in ensuring anonymity. These operations, while essential, add to the computational cost. The result is displayed in Fig. 6. The blinding operation involves multiplying the public key with a random blinding factor, while the unblinding operation removes the blinding factor from the signed message. The computational overhead of these operations is directly proportional to the size of the key and the blinding factor. In the proposed scheme, the blinding and unblinding operations are performed on the client-side, distributing the computational load and reducing the burden on the server. While this overhead is noticeable, it is still within acceptable limits and does not significantly degrade the overall performance of the authentication process.

C. Signature Verification

The task of the verifier, which involves checking the signature against the challenge, may introduce some latency, particularly when dealing with multiple requests concurrently. However, this step can be efficiently performed on high-performance servers without requiring a secure computing environment. A typical server has the capability to complete over 20,000 signature verification operations per second, making it a non-bottleneck. The signature verification process in the proposed scheme remains identical to the standard FIDO2 protocol, ensuring compatibility and leveraging the existing infrastructure. The high throughput of signature verification

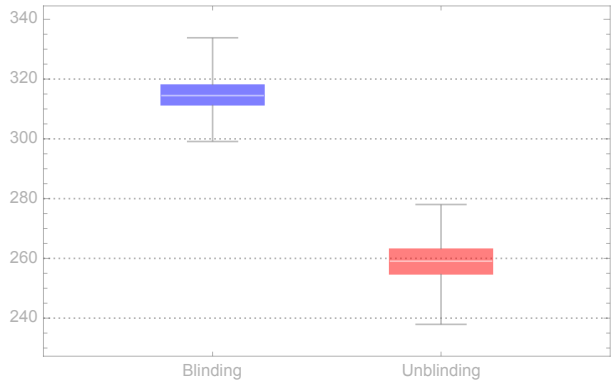

Fig. 6. Comparison of the time taken for blinding and unblinding operations with standard FIDO2 operations

on the server-side ensures that the proposed scheme can scale well and handle a large number of concurrent authentication requests without introducing significant latency.

D. Communication Overhead

The proposed design introduces additional communication steps between the user, the verifier, and the relying party. Evaluating the data exchanged during these interactions provides insights into potential bandwidth costs. The communication overhead arises from the exchange of blinded public keys, signed challenges, and unblinded signatures. However, the size of these additional data elements is relatively small compared to the overall communication in the FIDO2 protocol. The blinded public key and the signed challenge are typically a few hundred bytes each, while the unblinded signature is similar in size to a standard FIDO2 signature. Moreover, the communication overhead is mitigated by the fact that the proposed scheme does not introduce any additional round trips or protocol messages. The additional data elements are piggy-backed on the existing FIDO2 messages, minimizing the impact on network latency and bandwidth consumption.

In summary, the computational and communication overhead introduced by the *anon-auth* design is manageable and does not significantly degrade the performance or user experience of the FIDO2 protocol. The enrollment process, while more time-consuming than the standard FIDO2 enrollment, is a one-time setup and can be performed offline. The blinding and unblinding operations add a small computational overhead on the client-side, but this overhead is distributed and does not burden the server. The signature verification process remains efficient and scalable, leveraging the existing FIDO2 infrastructure. The communication overhead is minimal and does not introduce additional round trips or protocol messages. Overall, the proposed scheme strikes a balance between enhancing privacy through anonymity and maintaining the performance and usability of the FIDO2 protocol.

VII. Conclusion

The proposed FIDO2 anonymous authentication extension presents a promising solution to the challenges of integrating anonymity into existing authentication protocols. While the design introduces computational and communication overheads, the benefits of enhanced user privacy and security are evident. However, as with any novel approach, continuous evaluation and refinement are essential to ensure its viability in real-world scenarios. By providing a comprehensive evaluation framework, this section aims to offer a clear understanding of the costs associated with the proposed design. As the digital world continues to evolve, the demand for secure and privacy-respecting solutions will only grow, making research in this domain increasingly relevant.

Acknowledgment

This work is supported in part by the National Key R&D Program of China under grant No. 2021YFB2900100, the National Natural Science Foundation of China (NSFC) under grants No. 62302259.

References

[1] K. Bicakci and Y. Uzunay, "Is FIDO2 Passwordless Authentication a Hype or for Real? A Position Paper," in *Proceedings of the 15th International Conference on Information Security and Cryptography (ISCTURKEY)*, 2022, pp. 68–73.

[2] "Yubico — yubikey strong two factor authentication," https://www.yubico.com/, (Accessed on 09/30/2024).

[3] "Canokeys," https://www.canokeys.org/, (Accessed on 09/30/2024).

[4] "What is a passkey? — passkey.org," https://passkey.org/, (Accessed on 09/30/2024).

[5] N. Asghar, "A survey on blind digital signatures," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:44319470

[6] S. Ghorbani Lyastani, M. Schilling, M. Neumayr, M. Backes, and S. Bugiel, "Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication," in *Proceedings of the 2020 IEEE Symposium on Security and Privacy*, 2020, pp. 268–285.

[7] J. Kunke, S. Wiefling, M. Ullmann, and L. L. Iacono, "Evaluation of Account Recovery Strategies with FIDO2-based Passwordless Authentication," 2021.

[8] E. Klieme, J. Wilke, N. van Dornick, and C. Meinel, "FIDOnuous: A FIDO2/WebAuthn Extension to Support Continuous Web Authentication," in *Proceedings of the 19th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2020, pp. 1857–1867.

[9] L. Würsching, F. Putz, S. Haesler, and M. Hollick, "FIDO2 the Rescue? Platform vs. Roaming Authentication on Smartphones," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23, 2023.

[10] D. Chaum, "Blind signatures for untraceable payments," in *Proceedings of CRYPTO'82*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds., 1983, pp. 199–203.

[11] J. L. Camenisch, J.-M. Piveteau, and M. A. Stadler, "Blind signatures based on the discrete logarithm problem," in *Proceedings of EUROCRYPT'94*, A. De Santis, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 428–432.

[12] M. Rückert, "Lattice-based blind signatures," in *Proceedings of ASIACRYPT 2010*, M. Abe, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 413–430.

[13] S. Ibrahim, M. Kamat, M. Salleh, and S. Aziz, "Secure E-voting with blind signature," in *Proceedings of the 4th National Conference of Telecommunication Technology*, 2003, pp. 193–197.

[14] I. Teranishi, J. Furukawa, and K. Sako, "k-Times Anonymous Authentication (Extended Abstract)," in *Proceedings of ASIACRYPT 2004*, P. J. Lee, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 308–322.

[15] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, "How to Win the Clonewars: Efficient Periodic n-Times Anonymous Authentication," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06, 2006, p. 201–210.

[16] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

[17] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 384–394, 2014.

[18] F. Dang, P. Zhou, Z. Li, E. Zhai, A. Mohaisen, Q. Wen, and M. Li, "Large-scale invisible attack on afc systems with nfc-equipped smartphones," in *Proceedings of the 2017 IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[19] F. Dang, P. Zhou, Z. Li, and Y. Liu, "Nfc-enabled attack on cyber physical systems: A practical case study," in *Proceedings of the 2017 IEEE Conference on Computer Communications Workshops*, 2017, pp. 289–294.

[20] M. Kepkowski, M. Machulak, I. Wood, and D. Kaafar, "Challenges with Passwordless FIDO2 in an Enterprise Setting: A Usability Study," 2023.

[21] L. Hanzlik, J. Loss, and B. Wagner, "Token meets Wallet: Formalizing Privacy and Revocation for FIDO2," in *Proceedings of the 2023 IEEE Symposium on Security and Privacy*, 2023, pp. 1491–1508.

[22] L. Lassak, A. Hildebrandt, M. Golla, and B. Ur, "It's Stored, Hopefully, on an Encrypted Server: Mitigating Users' Misconceptions About FIDO2 Biometric WebAuthn," in *Proceedings of the 30th USENIX Security Symposium*. USENIX Association, Aug. 2021, pp. 91–108. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/lassak

[23] "canokeys/canokey-core: Core implementations of an open-source secure key," https://github.com/canokeys/canokey-core/, (Accessed on 09/20/2023).

[24] "Yubico/libfido2: Provides library functionality for fido2, including communication with a device over usb or nfc." https://github.com/Yubico/libfido2, (Accessed on 09/20/2023).

[25] F. Dang, L. Li, and J. Chen, "xRSA: Construct Larger Bits RSA on Low-Cost Devices," in *Proceedings of the 27th International Conference on Parallel and Distributed Systems*, 2021, pp. 637–643.