Authentication Module Based on the Protocol of Zero-Knowledge Proof*

Alexander M. Kadan[1][0000-0003-3701-8100], Egor R. Kirichonok[2][0000-0002-5904-6391]

[1] Yanka Kupala State University of Grodno, Grodno, Belarus,
kadan@mf.grsu.by
[2] Yanka Kupala State University of Grodno, Grodno, Belarus,
kirichonok_er_17@mf.grsu.by

Abstract. This article discusses passwordless authentication methods. These methods are now becoming commonplace and eliminate the problems associated with the difficulty of remembering secrets. Passwordless authentication has clear security and privacy advantages over traditional authentication methods. The usability of passwordless authentication depends on the type of authenticator used. The paper proposes an implementation of an authentication module based on the Zero Knowledge Proof (ZKP) protocol. The issues of its application for passwordless user authentication to web application resources are discussed within the framework of the Web Authentication (WebAuthn) passwordless web authentication standard developed by the FIDO Alliance. The module is based on the use of the FIDO2 authenticator. It also allows the use of various authenticators, including hardware keys connected to the device via USB, Bluetooth Low Energy or NFC, software keys, the implementation of which can be very different. Currently, the cost of implementing passwordless authentication can be significant. This is a major obstacle to the widespread adoption of this advanced technology.

Keywords: Zero-Knowledge Proof, ZKP, Cryptographic Protocols, Authentication, Web Authentication, WebAuthn, FIDO Alliance.

1 Introduction

In cryptography, Zero-Knowledge Proof (ZKP) is considered as an interactive protocol that allows one of the parties (the Verifier) to verify the validity of a statement (usually mathematical) without receiving any other information from the second party (the Prover), neither the content of the statement nor the source from which the Prover learned about its truth.

In particular, ZKP can play the role of a tool that verifies data and users, granting privileged access and establishing trusted connections. For example, one of the obvious applications of the ZKP protocol is checking whether a user has certain permissions

when requesting access to information system resources, without disclosing the details of these permissions to the protocol participants. This protocol can also be used in tasks where it is necessary to ensure the security of data (for example, personal information) during financial transactions. Since the use of the ZKP protocol allows you to reliably protect the user's data, due to the absence of their storage in the system and transmission over network communications.

In this work, along with the study of the general theoretical aspects of the use of ZKP methods, it is also expected to study aspects of the use of ZKP in the design and development of applied solutions. The task of developing a prototype of the application system and evaluating the effectiveness of using the authentication method based on ZKP is set and solved.

2 Key features of ZKP protocols

Zero-knowledge proofs are not proofs in the mathematical sense of the term, because there is some small probability that the Prover will be able to trick the Verifier into a false statement (correctness error). In other words, zero-knowledge proofs are probabilistic proofs, not deterministic ones. Nevertheless, there are methods to reduce the correctness error to negligible values [1, 5].

Zero-knowledge proof protocols must have three properties:

1. Completeness: if the statement is true, then the Prover will convince the Verifier of this with any predetermined accuracy.
2. Correctness: if the statement is false, then any, even “dishonest” Prover will not be able to convince the Verifier except for a negligible probability.
3. Zero-knowledge: if the statement is true, then any, even "dishonest" Verifier will not know anything except the very fact that the statement is true [4].

The interactivity of the protocol means the direct exchange of information between the parties [1]. The traditional ZKP protocol requires interactive input from the Verifier, usually in the form of a task or problem. The goal of the legal Prover (who has proof) in this protocol is to convince the Verifier that he has a solution, without giving away even part of the “secret” proof (“zero-knowledge”). The purpose of the Verifier is to make sure that the Prover “doesn't lie” [1].

Each round, or proof accreditation, consists of three stages. They can be schematically depicted as follows (here P is the Prover, V is the Verifier):

— P => V: witness
— P <= V: challenge
— P => V: response

First, P selects from a predetermined non-empty set some element, which becomes its secret – a private key. The public key is calculated from this element and then published. Knowing the secret determines the set of questions to which P can always give correct answers. Then P selects a random element from the set, computes the proof according to certain rules (depending on the specific algorithm), and then sends it to V.

After that, V selects one of the whole set of questions and asks P to answer it (challenge). Depending on the question, P sends a V answer [6]. The information V received is enough to check whether P owns the secret. The rounds can be repeated as many times as you like until the probability that P "guesses" the answers is low enough. This approach is also called "cut-and-choose", first used in cryptography by Michael Rabin [1, 7, 14].

Zero-knowledge proof protocols were also developed [2, 3], which did not require the presence of interactive source data, while the proof of which, as a rule, relies on the assumption of an ideal cryptographic hash function, that is, it is assumed that the output of a one-way hash function cannot be predicted if its input is not known [1].

3 New standards for passwordless authentication

In the practical field, technologies such as single sign-on and two-factor authentication are widely used to protect web applications. Recently, there has been considerable interest in a new (and often misunderstood) trend known as passwordless web authentication [8, 12, 13, 15-17].

A common property of passwordless authentication schemes is that they do not require a password in the traditional sense.

The concept of a multi-factor authenticator (what you have) is activated by either a PIN (something you know) or biometric (what you have). Multifactor Authenticator provides multifactor authentication without stacking one-factor authenticators on top of each other.

Let's briefly dwell on the concept of the FIDO2 authenticator, that is, a multi-factor cryptographic authenticator that is compatible with the W3C Web Authentication (WebAuthn) standard.

The FIDO Universal 2nd Factor (U2F) authentication protocol was the starting point for the FIDO2 project, a joint project of the World Wide Web Consortium (W3C) and the FIDO Alliance. The results of the project include the W3C Web Authentication (WebAuthn) standard [9] and the FIDO Client to Authenticator Protocol (CTAP) specification [10].

Together, WebAuthn and CTAP are known as FIDO2, a generic term for one (or both) of these technology standards.

The FIDO2 authenticator, also called the WebAuthn authenticator, uses public-key cryptography to communicate with the WebAuthn client. A type of FIDO2 authenticator, called a platform authenticator, is tightly integrated into the WebAuthn client platform, that is, implemented on the client device itself.

The FIDO2 authenticator can be used in both single-factor and multi-factor modes. In one-factor mode, the authenticator is activated like any other one-factor authenticator, that is, by a simple test of the user's presence (for example, pressing a button). In the multi-factor model, the authenticator (what you have) is activated by either a PIN (what you know) or biometric (what you are).

Client to Authenticator Protocol (CTAP) allows an authenticator (such as a hardware security key) to communicate with a client platform (such as a laptop). The CTAP-

compliant authenticator connects to the client through one or more of the following transport bindings: USB, NFC, or Bluetooth Low Energy (BLE).

In March 2019, a standard dedicated to passwordless web authentication – Web Authentication (WebAuthn) was presented to the public. The standard was developed by the FIDO Alliance, which aims to develop authentication standards that do not rely on passwords. On March 4, 2019, the standard was recommended for use by the international organization World Wide Web Consortium, which deals with the issues of Internet standards [11].

The W3C Web Authentication (WebAuthn) standard is the centerpiece of the FIDO2 project. WebAuthn includes the website, web browser, and authenticator:

- The website is the respective relying party of WebAuthn.
- The browser is a WebAuthn compatible client
- The authenticator is the corresponding WebAuthn Authenticator (also called FIDO2 Authenticator).

WebAuthn indicates how the applicant demonstrates ownership and control of the WebAuthn authenticator to a verifier called the WebAuthn relying party. The authentication process is done through an object called a WebAuthn client, which is nothing more than a corresponding web browser [11].

4 Using ZKP Protocols for Authentication

Zero-knowledge proof research is motivated by authentication systems in which one party wants to prove its identity to another party using some secret information (such as a password), but does not want the other party to know anything about the secret. This is called " zero-knowledge proof of knowledge". However, the password is usually too small or not random enough to be used in many zero-knowledge-proof schemes. Zero-knowledge password confirmation is a special type of zero-knowledge confirmation of knowledge that deals with the limited size of passwords.

When a user logs in to the passwordless authentication system's authentication server, some mathematical problems are sent to their browser from the server, which requires answers. Authentication is only allowed when the user's browser responds correctly to all calls. For each new verification attempt, a different set of problems is presented.

5 Cryptographic protocols used by WebAuthn

Thanks to the WebAuthn standard, it became possible to use many different options for authentication (see, for example, Fig. 1), including hardware keys connected to a device via USB, Bluetooth Low Energy or Near-Field Communications (NFC); software keys, the implementation of which can be very different. The vast majority of organizations

that need strong authentication are interested in using both zero-knowledge proof protocols (to protect their resources in general) and the WebAuthn standard (to protect web resources).

WebAuthn standardizes the interaction of a website, web browser, and authenticator [8]:

- The website is the relying party of WebAuthn.
- Browser is a WebAuthn client.
- The Authenticator is a FIDO2 Authenticator, which means it is assumed to be compatible with the WebAuthn client.

WebAuthn defines how a client proves its identity to a verifier, called the WebAuthn Relying Party.

In any case, the client proves that he has a FIDO2 authenticator (something we have). Depending on the type of authenticator, the following authentication factors can also be used (additionally):

- something that we know (password, pin-code, graphic).
- something that is an integral part of ourselves (such as a voice or a fingerprint).

Fig. 1. Hardware keys: with a button (left) and with a fingerprint scanner (right)

6 The prototype of a Resource Access System Based on ZKP Protocols

The structurally designed software consists of two parts, a web application and a server responsible for handling REST requests. The web application is implemented using the following technologies: JavaScript programming language; Ionic framework; Angular framework. The RESTful server is implemented using the following technologies: Java programming language; Spring Boot framework.

For the system to work on the client-side, the following is required:

- Availability of FIDO2 hardware or software authenticator. For example Android platform (from version 7), Windows 10 OS, hardware key (for example, Yubico or Feitian).
- The browser that supports the Web Authentication standard. For example Microsoft Edge, Mozilla Firefox, Google Chrome, Apple Safari, Opera Web Browser, iOS Safari, Android Browser, Chrome for Android, Firefox for Android.
- Access to the Internet or a local network on one of the nodes of which the server is deployed.

For the system to work from the server-side:

- Installed database management system (DBMS) MariaDB.
- Installed Java Runtime Environment (JRE).
- Certificate for the site (HTTPS protocol), since the Web Authentication standard does not work over the HTTP protocol (except for localhost - the same computer on which the server is running).
- Software platform Node.js.
- Access to the Internet or access to the local network if the application is supposed to be deployed locally.

The sequence diagram of the registration process is shown in Figure 2. It should be noted that any unauthenticated request other than registration and authentication requests is denied and will be redirected to the authentication page. From this page, you can go to the registration page. You can also go to the registration page by making an HTTPS get-request to the address HTTPS://{server IP or its domain name}/#/registration. To register, you must enter a username. After that, the FIDO2 authenticator will ask the user for confirmation, if necessary (some authenticators do not need confirmation, the very fact of the presence of an authenticator is enough). After successful confirmation, the user will be redirected to the registration end page, where he will receive a recovery code (a public key variant), which can be used to confirm his identity and change the authenticator (in case of loss of the authenticator). This code should be kept in a safe place. After that, the registration process is considered complete (Fig. 2).

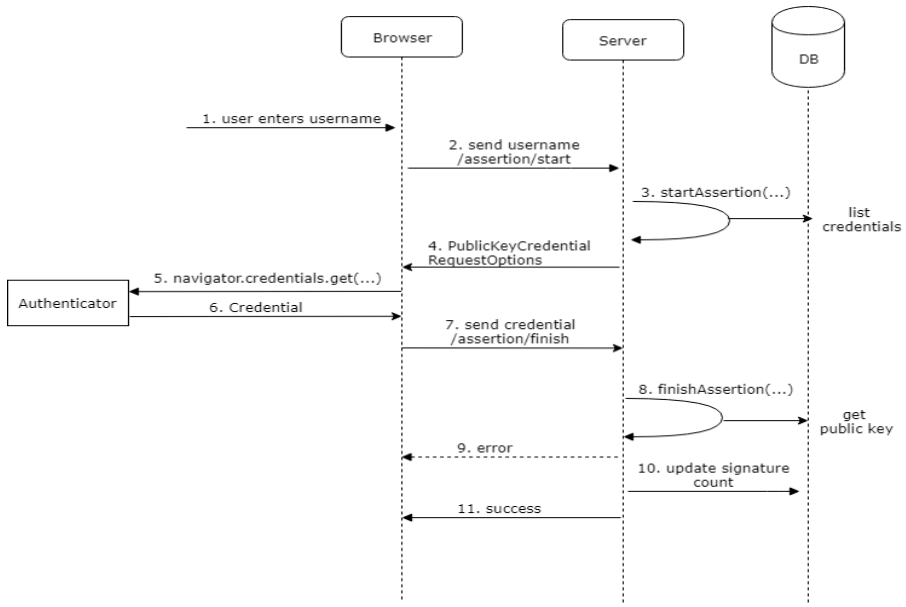

Fig. 2. Registration: sequence diagram

The sequence diagram of the authentication process is shown in Figure 3.

The client-side authentication process is similar to the registration process: you need to enter the username and verify the identity for the FIDO2 authenticator, if necessary.

After successfully verifying the identity of the authenticator, the user will be redirected to the home page. On this page, he gets the opportunity to log out of the system, as well as start registering an additional authenticator (this functionality will undoubtedly be needed if the user needs to log in under the same account from different devices).

In addition to direct registration and authentication, the application has such features as registering an additional authenticator and replacing the authenticator.

To register an additional authenticator, the user must enter a special code that is generated by the system and confirm his identity again.

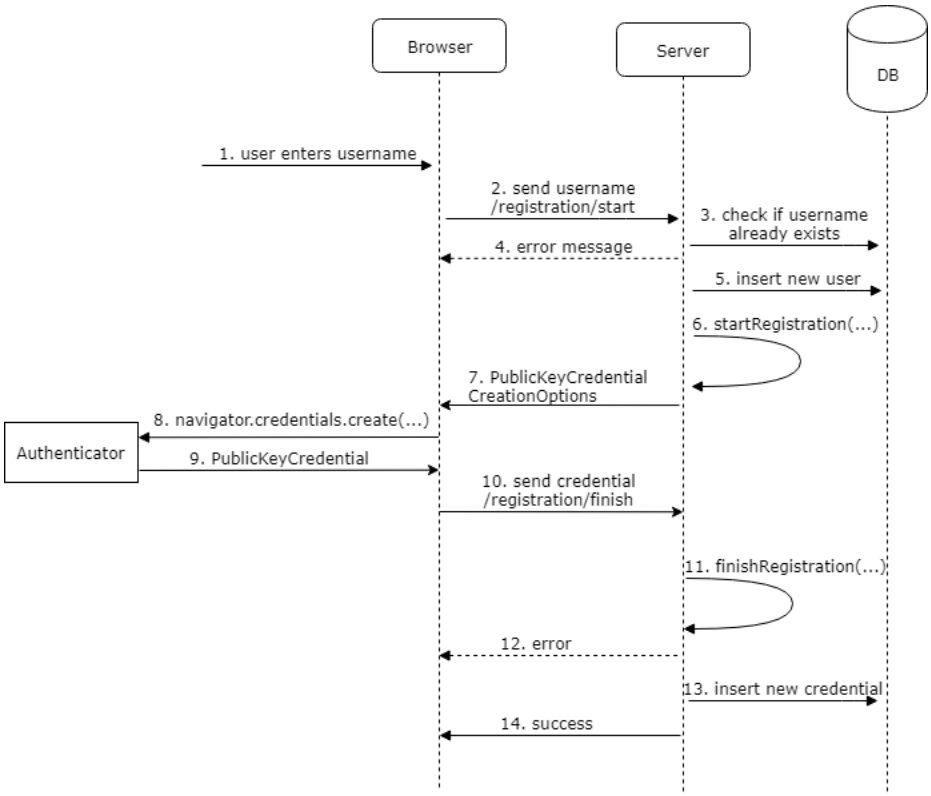

Fig. 3. Authentication: sequence diagram

To replace the authenticator, you must use the recovery code received by the user during registration. This code must be entered on the registration page, after which the identity of the new authenticator must be verified. This functionality is especially necessary if the authenticator is lost or stolen.

After confirming the identity, the user goes to the registration completion page, as when the first registered and receives a new recovery code.

7 Conclusions

In this work, within the framework of the ZKP methods and the WebAuthn standard, the registration and authentication processes were considered in detail. Several zero-knowledge-proof protocols can be used by the WebAuthn standard.

The practical part of the work is the designed and implemented the system for accessing web resources based on zero-knowledge-proof protocols. The functionality of the developed system includes the functions of registration, authentication, registration of an additional authenticator, replacement of the authenticator.

The direction of further research may be the development of your software authenticator. However, the expediency of this is still in doubt.

References

1. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. Published by Wiley, Second Edition, November. 784 pages. (1995)
2. De Santis A., Micali, S., Persiano, G.: Non-Interactive Zero-Knowledge Proof Systems. In: Pomerance C. (eds) Advances in Cryptology — CRYPTO ’87. CRYPTO 1987. Lecture Notes in Computer Science, vol 293. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48184-2_5 (1988)
3. Blum, M., Feldman, P., Micali, S.: Non-Interactive Zero-Knowledge and its Applications. STOC'88: Proceedings of the twentieth annual ACM symposium on Theory of computing New York City: ACM. P. 103–112. doi:10.1145/62212.62222 (1988).
4. Menezes, A. J., Oorschot, P. V., Vanstone, S. A.: Chapter 10. Handbook of Applied Cryptography – CRC Press, P. 405–417. (1996).
5. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. M. Sudan – Society for Industrial and Applied Mathematics, Vol. 18, Iss. 1. P. 186–208. doi:10.1137/0218012 (1989).
6. Mao, V.: Modern cryptography: Theory and practices. Moscow: Williams, 2005. 768 p.
7. Rabin, M.O.: Digital Signatures. Foundations of Secure Computation. New York: Academic Press, C. 155—168. (1978).
8. Grassi, Paul A., Garcia, Michael E., Fenton, James L.: NIST Special Publication 800-63-3: Digital Identity Guidelines. National Institute of Standards and Technology (NIST). doi:10.6028/NIST.SP.800-63-3. (June 2017).
9. Balfanz, D., Czeskis, A., Hodges, J., Jones, J.C., Jones, M. B., Kumar, A., Liao, A., Lindemann, R., Lundberg, E., eds.: Web Authentication: An API for accessing Public Key Credentials Level 1 (Recommendation ed.). World Wide Web Consortium (W3C). Retrieved 4 March 2019. https://www.w3.org/TR/webauthn-1/ (4 March 2019).
10. Brand, Ch., Czeskis, A., Ehrensvärd, J., Jones, M. B., Kumar, A., Lindemann, R., Powers, A., Verrept, J., eds. Client to Authenticator Protocol (CTAP). FIDO Alliance. Retrieved 7 March 2019 https://fidoalliance.org/specs/fido-v2.0-ps-20190130/fido-client-to-authenticator-protocol-v2.0-ps-20190130.html (January 30, 2019).
11. Web Authentication: An API for accessing Public Key Credentials Level 1. URL: https://www.w3.org/TR/webauthn-1/. Last access: 16.04.2020.

12. Khernane, N., Potop-Butucaru, M. and Chaudet, C.: BANZKP: A Secure Authentication Scheme Using Zero-Knowledge Proof for WBANs, 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Brasilia, pp. 307-315, DOI: 10.1109/MASS.2016.046. (2016)
13. De Santis, A., and Persiano, G.: Zero-Knowledge Proofs of Knowledge Without Interaction. Proceedings., 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, PA, USA, pp. 427-436, DOI: 10.1109/SFCS.1992.267809. (1992)
14. Merkle, R.C.: A Certified Digital Signature. In: Brassard G. (eds) Advances in Cryptology — CRYPTO' 89 Proceedings. CRYPTO 1989. Lecture Notes in Computer Science, vol 435. Springer, New York, NY. https://doi.org/10.1007/0-387-34805-0_21 (1990)
15. Guirat, I. B., and Harry H.: Formal Verification of the W3C Web Authentication Protocol. In Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security (HoTSoS '18). Association for Computing Machinery, New York, NY, USA, Article 6, 1–10. DOI:https://doi.org/10.1145/3190619.3190640 (2018).
16. Bonneau, Joseph; Herley, Cormac; Oorschot, Paul C. van; Stajano, Frank: The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. University of Cambridge Computer Laboratory, Technical Report Number 817. Cambridge, UK. ISSN 1476-2986. (2012).
17. Grassi, Paul A.; Garcia, Michael E.; Fenton, James L. "NIST Special Publication 800-63-3: Digital Identity Guidelines. National Institute of Standards and Technology (NIST). DOI:10.6028/NIST.SP.800-63-3. (June 2017).