


Article

Securing Remote Access to Information Systems of Critical Infrastructure Using Two-Factor Authentication

Rasa Bruzgiene *  and Konstantinas Jurgilas

Department of Computer Sciences, Kaunas University of Technology, Studentu Str. 50-211, 51368 Kaunas, Lithuania; konstantinas.jurgilas@ktu.edu

* Correspondence: rasa.bruzgiene@ktu.lt

Abstract: Information systems of critical infrastructure provide services on which the core functions of a state and its economy depend as well as welfare of society. Such systems are becoming an increasingly common target for crimes and attacks in cyberspace, as their vulnerabilities can be exploited for malicious activities seeking financial or political gain. One of the main reasons that threatens the security of these systems is the weak control of remote access, otherwise defined as management of a system's user identity. Management of user identity depends on user authentication, authorization and the assignment of certain rights in the digital space. This paper provides the proposed two-factor (2FA) digital authentication method for remote access to an information system of a critical infrastructure. Results of testing the method's usability and resilience to cyber threats have shown that the system, in which the method was implemented, is protected from dangerous HTTP requests and publicly available system's endpoints are protected from threatening inputs that could cause malicious activities on the critical infrastructure. Additionally, the implementation of the authentication API application ensures the rapidity of the method for less than 500 ms for 100 users working in parallel with the system at the same time.



Citation: Bruzgiene, R.; Jurgilas, K. Securing Remote Access to Information Systems of Critical Infrastructure Using Two-Factor Authentication. *Electronics* **2021**, *10*, 1819. <https://doi.org/10.3390/electronics10151819>

Academic Editor: Lei Shu

Received: 2 July 2021

Accepted: 26 July 2021

Published: 29 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: 2FA authentication; critical infrastructure; penetration testing; vulnerability; cyber-attack

1. Introduction

Information systems of critical infrastructure are consisting of hardware and software that together provide certain core functions and services the disruption of which could cause significant damage to national security, stability of economics or human health and well-being [1]. Such systems are used in the sectors of energy, information and communication technology, water supply, health, finance, transport, chemical, research of space, national security and etc. The information systems of critical infrastructure are very important as they ensure the continuity of services provided in those sectors. In this regard, such systems sometimes are referred also as critical systems.

Society is dependent on the provision of these services and any disruption to it can result failures of services ranging from short-term disruptions (for example, interruption of electricity or water supply) to serious and dangerous disasters impacting whole operation of critical infrastructure. For instance, a malfunctioning of nuclear power plant can cause an explosion that would result in the release of radioactive particles into the atmosphere. In this case it would be a disaster to whole world. A malfunctioning of the systems in critical infrastructure can be implied by various factors: natural disasters (e.g., earthquakes, tsunami, hurricanes, etc.), technology-related disruptions (e.g., software failures, power blackout, etc.) and malicious activities by humans, basically expressed as attacks in cyberspace.

Historically, information systems of the critical infrastructure often have been designed to operate only on internal networks with isolation from public networks. However, the need to share information with different units in business and at the same time to have

remote access to critical systems have led to the connection of critical infrastructure to a public Internet network. Over the past several years, there is a noticeable trend that more and more information systems of the critical infrastructures are being moved from on-premise to the clouds. The benefits of cloud computing are weighty to the operation of critical systems. Redundancy, reliability, availability, and scalability are very important features to critical systems provided by cloud computing. However, the transition of critical systems to cloud computing opens not only benefits, but challenges and issues as well [2]. The operation of the information systems of critical infrastructure in the wider cyberspace is increasingly becoming a target for cyber criminals, which in turn poses significant threats to the security of such systems.

In 2019, warnings that cyber threats pose a risk to public welfare, security and prosperity were published in National Intelligence Strategy Report of United States [3]. Those warnings were related with the fact that information technologies are inseparable from critical infrastructures and widely used by society [1]. The report also sets out the objectives for a national security, one of which is to protect critical systems.

It is essential to note that attacking of critical infrastructures in cyberspace already has been performed a good decade ago. In 2010, a nuclear fuel plant was infected with the Stuxnet virus in Natanz, Iran. This virus was created in order to damage the control engines commonly used in centrifuges of uranium enrichment. The virus has managed to shut down temporarily 1000 centrifuges at the same time forcing to shut down the whole plant [1]. In 2015, Ukraine suffered a cyber-attack against the network for electricity supply. The cyber criminals broke into three companies of energy supply and turned off their power generators in three regions of Ukraine. That attack affected nearly a quarter of a million people who were left without electricity for six hours in the middle of winter. The investigation revealed that those criminals had used the BlackEnergy3 virus, which was allegedly distributed through the use of electronic phishing.

It is also worth noting that launching a virus on critical infrastructure is already done in a later stage of a cyber-attack. Before that, cyber malevolent must hijack the access to the critical systems. In 2017, cyber terrorists hijacked a remote access to a workstation that was allegedly located in Saudi Arabia [1]. The terrorists used a new virus, nicknamed Triton, which made it possible to hijack a control of the security system of a power plant. Investigators say it was sabotage aimed to cause an explosion in power plant by disabling a security system, which, in fact, was designed against catastrophic incidents. However, only a programming error of cyber terrorists led to the avoidance of the catastrophe. Evidence from the investigation of this cyber incident shows that the virus was spread through fraudulent emails.

Staying on the same note, the statistics of cyber security for past five years show a growing trend in the number of cyber-attacks against the systems of critical infrastructure as well in the detection of vulnerabilities and the gaps in security of such systems. The systems of energy, water supply and manufacturing of critical services have been identified as the most frequently attacked critical systems by Kaspersky report on security risks for industrial automation systems in 2018 [4]. The report states that the most common vulnerabilities detected on such critical systems are buffer overflows and improper validation of inputs. It is also stated that 16% of the detected vulnerabilities are related with the problems in authentication of system's users (incorrect authentication, authentication bypass, missing authentication in performing critical functions) and as well with the problems in access control (incorrect default permissions, incorrect management of privileges or incorrect management of authorization).

Inevitably, a remote access to the critical systems is the most common target for cyber criminals. When cyber criminals gain access to the information systems of the critical infrastructure, the consequences of attacks can cause a lot of damage as well can be very harmful to society. Poorly implemented solutions for remote access to the information systems of the critical infrastructure can lead to unauthorized access of cybercriminals whose primary purpose is to commit malicious, illegal activities. On the basis of that, one

of the most important security objectives is to ensure that only authorized persons can remotely access the systems of critical infrastructure and control the operation of it.

This paper presents the authors' proposed two-factor (2FA) digital authentication method for remote access to an information system of a critical infrastructure. The testing of the usability of 2FA digital authentication method as well as its resilience to cyber threats is provided and discussed also. The proposed method is designed according to the security requirements for the information systems of a critical infrastructure.

The scientific novelty and practicality of the proposed 2FA digital authentication method is based on the synergy between a "push notification" technology, digital certificates and authorization of the authentication requests in order to manage remote access to an information system of the critical infrastructure.

The remainder of this paper is structured as follows. After describing the relevance of this paper and security issues in critical infrastructures, Section 2 reviews the state-of-the-art in managing the identity of the entity, who is connecting remotely to critical infrastructure's systems. Section 3 describes the details of the 2FA digital authentication method, its operation processes and compliance with security requirements. Section 4 illustrates the testing of the method's usability and analysis of the results. For demonstration purposes, the example scenarios of cyber-attacks and its impact to the 2FA method-driven critical system is described in Section 5. Finally, concluding remarks and discussions are given in Section 6.

2. State-of-the-Art in Authentication Methods

Authentication is the process or the act of verifying and confirming the identity of an entity as the legal user of the system. The user must provide and prove his identity before using the system.

Basically, a certain identifier of a user is specified during the authentication process, e.g., username, employee code or email address. Then a user provides his private data for identity, e.g., password. If the provided private data is correct, the user's identity is confirmed and the user can access the system and successfully use the functions provided by it.

In order to ensure the higher reliability of the authentication process, it has been improved by a two-factor (2FA) authentication process [5]. During this process, the user must provide two different factors to verify his identity, such as providing a static login credentials and a token that was generated by a password generator. Besides 2FA authentication, a multi-factor (MFA) authentication can be used also. In this case, MFA authentication process is based on at least two authentication factors to confirm the identity of the user, who is accessing the critical system [6]. Generally, multi-factor authentication is designed on these core factors—personal data of the user (e.g., static login credentials), physical device or digital one-time data of the user (e.g., tablet, smart phone, short-time passcode) and biometric data of the user (e.g., fingerprints, voice patterns, arm geometry of the arm, etc.). However, multi-factor authentication not always means the more reliable process of the authentication. Especially, if the MFA is implemented in the information systems of the critical infrastructure. Particularly, such process needs more time than 2FA authorization and it is inconvenient in relation with critical systems. Secondly, it is hard to implement it across an entire critical system as well as integrate with other subsystems or systems driven by cloud computing. In the case of cloud computing, the dependence on third parties appears, when it comes to malfunctions occurred during the MFA authentication process. Thirdly, such authentication requires high resources for a maintenance. In response to this, 2FA authentication can serve as more flexible and at the same time reliable solution for the higher security of the critical infrastructure. Aside that 2FA authentication is a subset of MFA authentication, the reliability of it depends on how strong and reliable factors are integrated in it.

SMS messaging is one of the most common methods in two-factor authentication [7]. A user, who associates his login credentials with an active number of a mobile phone,

receives an SMS message with a passcode of 5 to 10 digits during the authentication process. The passcode must be entered into the login form to complete the authentication process successfully. However, SMS-based authentication method depends on the status of the mobile connection, i.e., without a mobile connection or during a travel abroad an SMS message with a passcode is potentially not sent [8]. If a mobile phone is lost with the SIM card, authentication is also not possible. Additionally, GSM communication is unreliable and leaking of SMS passcode by malicious software running on the mobile device can be a result of such communication. Moreover, the passcode is visible even if the phone is locked, in a case of the notification function of the lock window is on. SMS-driven authentication is vulnerable to social engineering attacks, where cybercriminals can obtain new SIM cards with an associated victim's mobile number. SMS messages will be sent to the malicious SIM card and the victim's mobile device will be disconnected from the mobile network. In the context of this, NIST announced [9] that SMS-based two-factor authentication methods are deprecated.

E-mail can also be used during the two-factor authentication process [10]. As with the SMS method, a one-time passcode of 5 to 10 digits is sent to the user's e-mail. The passcode can be presented to the user in several ways: as a text—the user has to copy the code to the login form or as a link—the user has to click on the link to complete the authentication process. However, such authentication has essential disadvantages: sometimes emails are not sent successfully, also the cybercriminals can hack account of a victim's email.

The authentication driven on one-time passwords is based on HMAC or Time-based One-Time Password (TOTP) [11] algorithms. These algorithms allow us to generate the passwords that are active for one use only or for a fixed period of time. The TOTP algorithm generates a one-time password using a specific cryptographic function that accepts a secret and a current time stamp as an input. SHA-256 is one of the cryptographic functions used for it. Notwithstanding that cryptographic functions are used such authentication has disadvantages also. First of all, a user cannot authenticate if the physical device, such as a smart phone, is turned off. Secondly, it is necessary to ensure the synchronization of the time between the mobile device and the system's server. Moreover, malicious people can generate passcodes themselves by taking over a shared secret as well as a brute force attack is possible when the number of access times is not checked.

Mobile devices can be used not only as one-time password or token generators during the two-factor authentication process, but also in the implementation of "*push authentication*" authentication method [12]. This method is based on the process when the users receive a message/request on the paired mobile device with which they can confirm or deny login to the account on a particular system [13]. It is worth noting that if a mobile device is stolen, the user must unlink the account from the stolen device as soon as possible. Moreover, the users can inadvertently accept illegal requests for login purposes as well as the login requests can be hijacked and automatically verified by malicious software running on the phone. In relation to this the secure and trust mobile application should be installed in the mobile device.

The location of a user can also be used during the authentication process. In [14] the location of the paired mobile device is checked during the authentication process. However, the location of a user is a private information that not every user wants to disclose. On the other hand, location information can be falsified at several levels - hardware, operating system or application levels [15].

Another way to authenticate a user is based on the recognition of the linked images [16]. Several images are presented to the user during such authentication and the user must select the images that were selected during the initial configuration of his login account. In practice, it takes time to the user to select images from a potentially long list of it. Besides that, there is always a possibility for a malicious hacker to select images based on a person's hobbies and psychological portrait.

Universal 2nd Factor (U2F) is an open standard that simplifies two-factor authentication by using specialized USB or NFC (Near Field Communication) devices and security

technologies of a smart card [17]. The U2F standard uses public key cryptography to authenticate a person, so the private key is always used only by a private person. U2F is a new technology that is not yet widely used, so a limited number of existing web browsers support it. The U2F standard is currently supported by Google Chrome, Opera and Firefox browsers. Internet Explorer and Edge browsers do not support it. Naturally, U2F-based devices are relatively expensive.

Biometrics is another solution to the authentication process [18]. A voice, a face, an eye iris, a hand geometry of a hand, fingerprints—a personal biometric data that can be used during the authentication process. Such biometric data is unique to each person and can be used to uniquely identify and authenticate a person. Different input devices are used for each biometric information along with different prices and, unfortunately, considerable costs. Consequently, the implementation of such a method itself is quite complex, expensive and requires additional hardware. It is important to note that the use of biometric data is protected by a law. On the other hand, the accuracy of the method is not 100%—a false rejection or confirmation of authentication is always possible. Moreover a legitimate user cannot change his biometric information and the compromise persists forever in a case of successful hijacking of such biometric data as well impersonating as another person.

Background monitoring by machine learning can be used to improve the authentication process. The implementation of machine learning methods makes it possible to monitor the behavior of the user and changes in it [19]. Such machine learning based authentication should allow us to determine when and how often the user of a system should authenticate himself. Sophisticated machine learning methods use a number of different behavioral factors for user authentication, such as the geographic location of the user, the time of day that the user tends to log in, how quickly the user can enter his password, keystroke dynamics, including mistakes that user can make i.e., accidentally entering an additional character in the password. However, it is unclear how many variables should be used by machine learning algorithms to properly and, what is even more important—reliably authenticate the user. Moreover, it is still quite challenging to make sure that machine learning algorithms will not analyse variables that are impossible to predict as such variables can be included entirely random and thus could lead to the emergence of a large number of false positives, that would result in the hijacked identity of the user by cyber criminals.

To summarize, passwordless authentication methods require more efforts to hijack a user's identity as cyber-attacks target into passwordless authenticators. However, even such methods still relatively depend on data of the user, whether biometric or user account data, when an account is being created or rebuilt. Such a viewpoint reminds that a balance in the usability, security and costs of implementation and operation of a particular authentication method should be maintained.

3. Proposed 2FA Authentication Method

The essential security requirements for a remote access to the information systems of a critical infrastructure must be ensured by these rules:

- the authentication of a remote user should be based on a strong multi-channel out-of-band authentication;
- a procedure of the authentication must be established in a such way that to enable the improvement of a request for remote access by a higher level employee in order to prevent remote access to critical infrastructure's systems by unauthorized persons;
- a processes for a remote access and authentication should be done over a secure channel (i.e., based on HTTPS protocol and VPN tunnels).

The proposed authentication method (see Figure 1) is based on these rules and follows a three-step authentication/authorization process. In order to gain a remote access to the information system of a critical infrastructure, a user needs:

1. to provide the account ID and a password, associated with it, that the user receives from the system's administrator during the creation of the user account. An authentication request will be sent to the user's mobile device in a case of provided correct data;
2. to approve successfully the authentication request, which was sent to the user himself, after that an authorization request will be send to the user's supervisor or several supervisors (as a higher level employee) asking if the applicant is allowed to connect to a particular system and whether remote access is allowed to him;
3. to connect to the critical system if one of the specified supervisors successfully approves the authorization request of a user and allows him a remote access to the secure system.

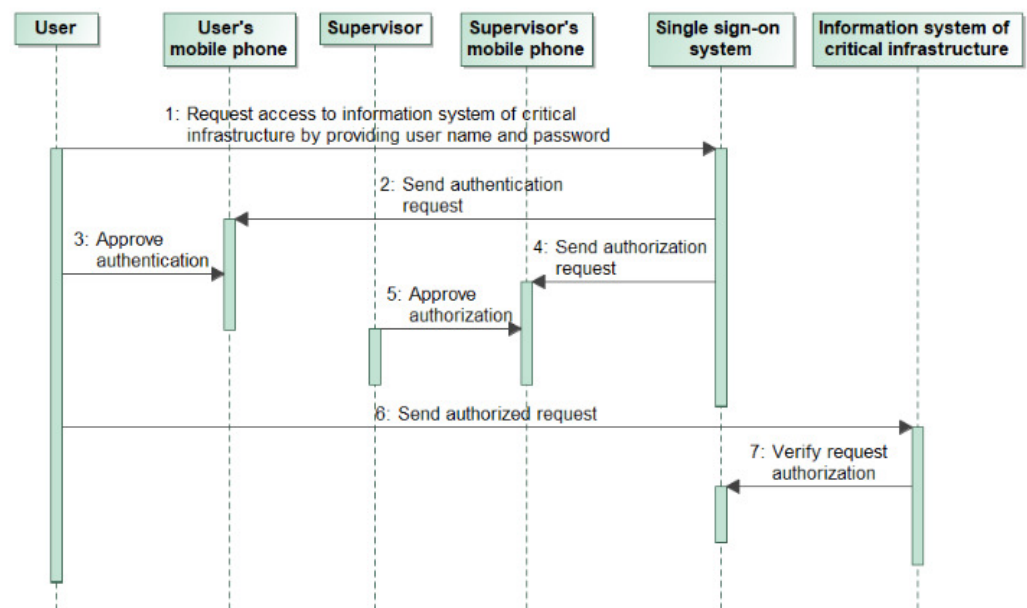


Figure 1. The proposed authentication method.

It is important to note here that the proposed authentication method is not applicable to the users who do not agree to provide their credentials during the authorization process. In this case, the user cannot log in as anonymous to a critical system. Additionally, the method is not applicable if the authorization request from a user cannot be validated by one of the supervisors when such higher-level employee simply do not exist. The system must provide the ability to specify several possible supervisors for the authorization of a system's user in order to increase redundancy in such cases when one of the supervisors will be unreachable due to any problems in a communication network. Following this idea, the implementation of the proposed digital authentication method would ensure to control a remote access to the information system of a critical infrastructure by the security measures set out in the security requirements for the critical infrastructure.

An algorithm for the intended operational process of allowing/rejecting a remote access to a system of a critical infrastructure is presented in Figure 2. This algorithm was compiled using the BPMN notation [20]. The positions of the employees according to the hierarchy of the organization—a user (i.e., an engineer) and a supervisor (i.e., a senior engineer)—are separated by the swimlanes.

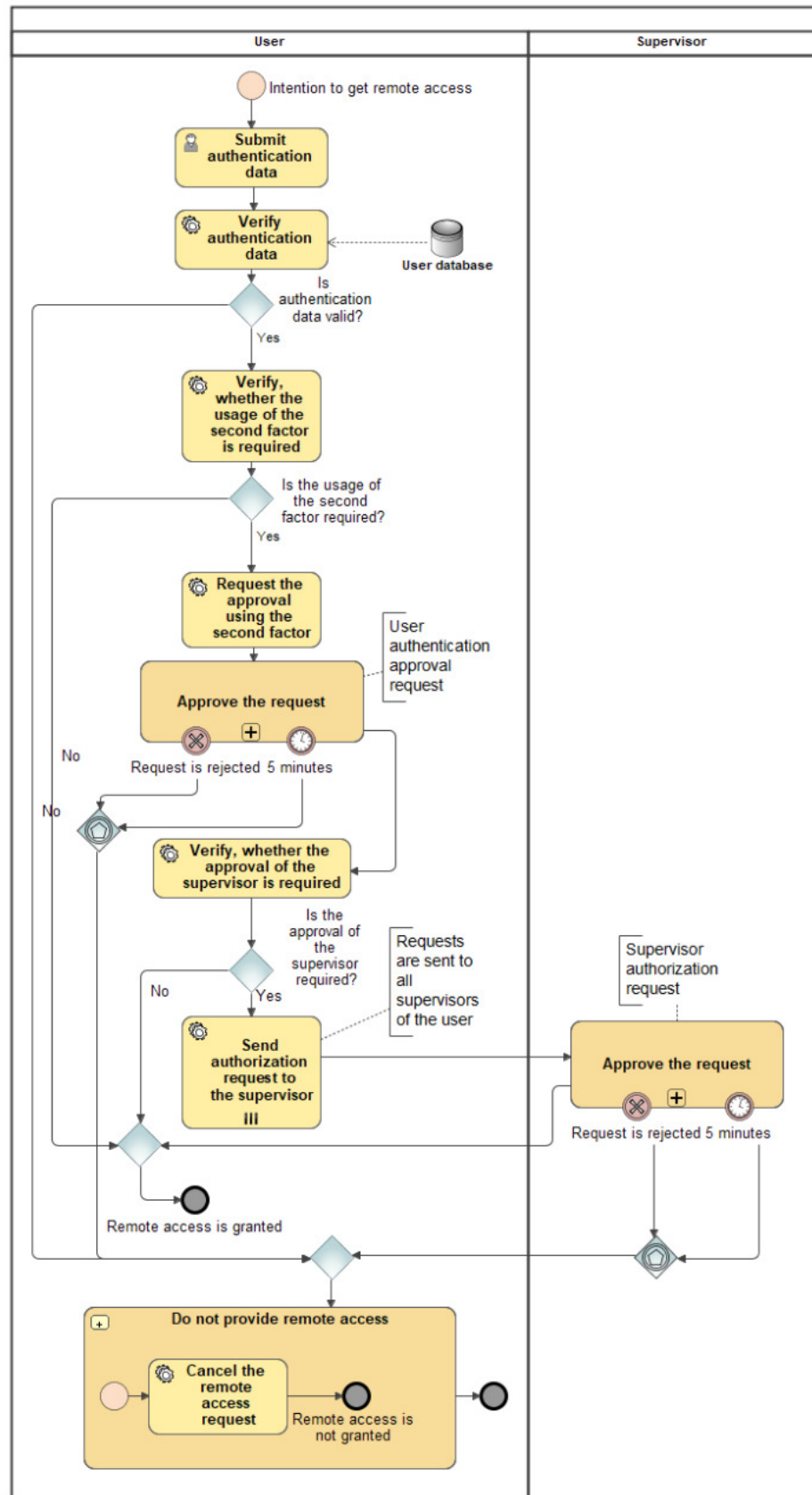


Figure 2. An algorithm for the intended operational process of allowing/rejecting a remote access to a system of a critical infrastructure (by BPMN).

The activity begins with the engineer’s (a user’s) desire to gain a remote access to a facility of a critical infrastructure. Naturally, a user needs to provide his credentials for authentication at first in order to verify it on the accounts’ database. The initiation of the request for a remote access is revoked if the credentials are incorrect. In this case the activity is terminated. Otherwise, checking whether a user has to confirm his identity using the second factor method or not begins. The user performs a sub-process of a request validation (Figure 3) if the use of the second factor for the authentication is necessary.

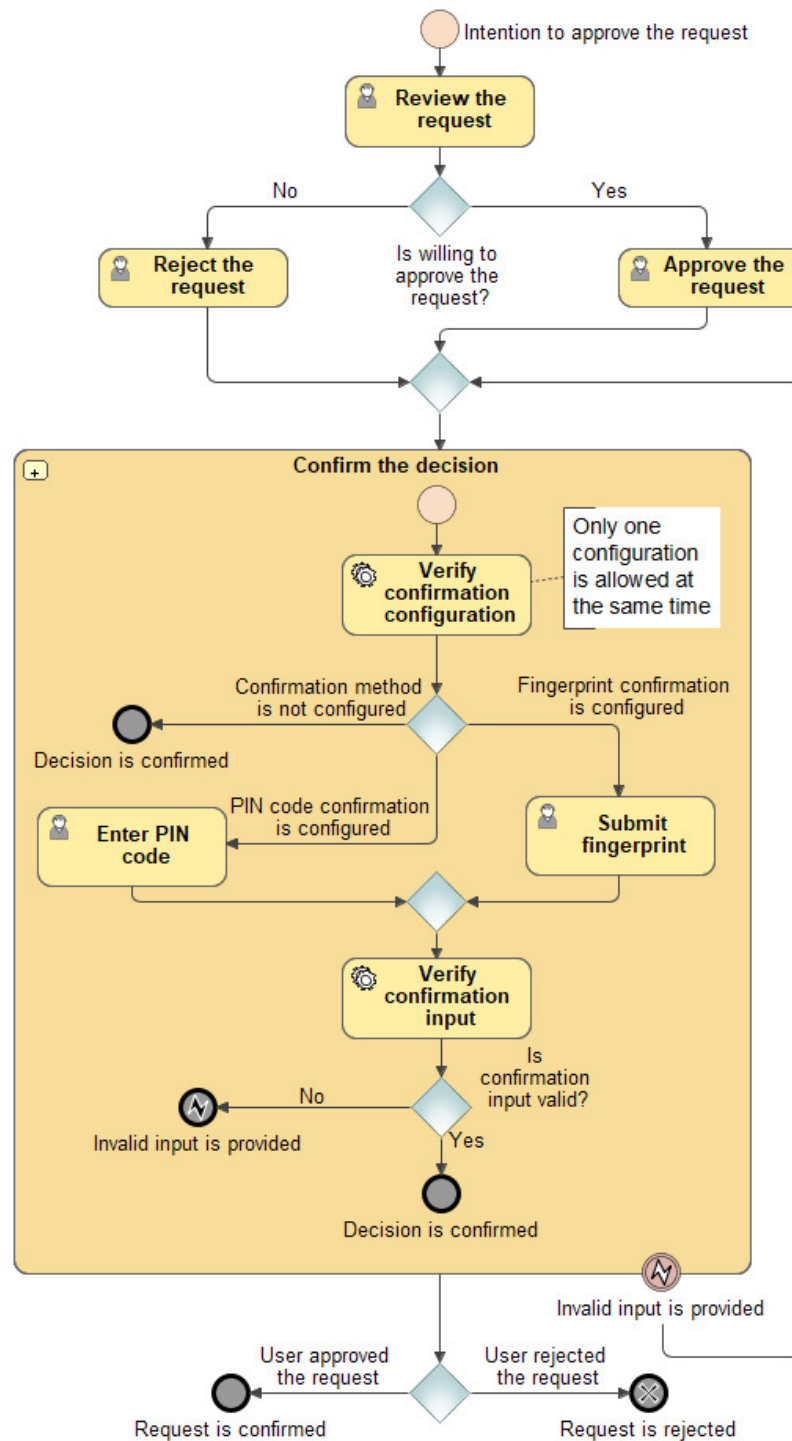


Figure 3. An algorithm for a process of a request validation (by BPMN).

Moving to the sub-process for the request validation, a user is presented with the request information that he needs to review. At this point, the user must decide whether to accept or reject the request. Checking whether the validation method is configured or not is done just after making this decision. In this case, the sequence of the actions performed by the user and supervisor is the same as it was mentioned earlier. The difference is in the content contained in the requests.

The same time-focused restrictions are applied to this sub-process. The request is canceled, and the activity is terminated without giving a remote access to a user in a case of rejected request or the excessive execution of the entire sub-process.

When a supervisor successfully approves a request of the user's authorization, then the activity is terminated by allowing the user to access the critical system remotely.

The proposed authentication method was evaluated according to the methodology for evaluating authentication schemes, which is described in [21]. The detailed description of this evaluation is provided in [22]. In category of the security 11 characteristics of the proposed method were evaluated:

1. Resilient-to-Physical-Observation,
2. Resilient-to-Targeted-Impersonation;
3. Resilient-to-Throttled-Guessing,
4. Resilient-to-Unthrottled-Guessing,
5. Resilient-to-Internal-Observation,
6. Resilient-to-Leaks-from-Other-Verifiers,
7. Resilient-to-Phishing,
8. Resilient-to-Theft,
9. No-Trusted-Third-Party,
10. Requiring-Explicit-Consent,
11. Unlinkable.

The evaluation of the characteristics belonging to the security category was performed using the numerical method of evaluation of qualitative characteristics. The results of the evaluation were compared with 8 other works [14,23–29]. The proposed method was evaluated in 10 points out of a maximum of 11, while couple of other works got maximum of 9.5 points. The security characteristic, named “No-Trusted-Third-Party” was rated for 0, as the single sign-on system implemented in the proposed method relies on the third-party Google Firebase service. In general, the proposed authentication method was evaluated with 24 points out of a maximum 30.5 and was evaluated as more appropriate in quality point of you in comparison with other works.

4. Testing the Usability of Proposed Method

Experimental testing of the proposed authentication method was carried out in different scenarios. In particular, testing of the method's usability according to its rapidity was carried out.

4.1. Testing Scenario

The prototype of a system, designed for the implementation of a proposed method, must follow one of the requirements—API access points (endpoints) need to process the requests from the users in less than 500 milliseconds, while the system is used by 100 users in parallel. The aim of the first experimental testing was to evaluate whether this requirement is met or not. Secondly, it was necessary to evaluate the response times of the authentication API access points and to investigate whether the API application can ensure stable operation in a case when specified number of parallel users loads the critical system.

The experiment was carried out using different volumes of data sets in order to evaluate the dependence of the method's rapidity on the amount of data as an additional load in the system. Data related with a real load on a system of a critical infrastructure is not available and open to the public, thus hypothetical data sets for experimental purposes

have been developed. The scenario for this experimental testing consists of several steps, the details of which are presented in Table 1.

Table 1. Phases of testing the rapidity of the proposed method.

Testing Phase	Endpoint	Application
Create authentication request	/api/oauth/request	Authentication API
Get authentication approval JWT token	/api/approvals/\${username}/authentication	Generation of data API
Approve authentication	/api/oauth/request/\${requestId}/approve-authentication	Authentication API
Get authorization approval JWT token	/api/approvals/\${username}/authorization	Generation of data API
Approve authorization	/api/oauth/request/\${requestId}/approve-authorization	Authentication API
Get authorization code	/api/oauth/request/\${requestId}	Authentication API
Exchange authorization code	/api/oauth/token	Authentication API
Renew access token	/api/oauth/token	Authentication API

4.2. Testing Setup

It is important to ensure that the environment for the actual use of the real critical system should be as close as possible during the performance of the experiments, therefore an automated tool was created that allows filling the database with the specified amounts of generated data.

A separate application, as an automated tool, based on a Spring Boot framework in Java programming language was written. The application for data generation provides the following functionality:

- the configuration file specifies how many accounts of the users as well OAuth 2.0 clients and historical authentication requests need to be generated;
- the specified amounts of data are generated and stored in the database when the application is launched;
- when the application is launched, there are two API access points for generating tokens in order to approve the requests for user's authentication and authorization. These API access points are designed to dynamically generate data generated by a mobile application.

JMeter tool, which executed HTTP requests to API access points in parallel, was used during the experiment. Management of the parallelism was based on the Ultimate Thread Group plugin, which allowed to specify how many parallel scripts should run, what the ramp-up and ramp-down periods are, and how long the experiment should take.

Configurations of data sets for the method's rapidity experiment are presented in Table 2.

Table 2. Configurations of data sets for the method's rapidity experiment.

Number of	#1	#2	#3	#4	#5
Parallel users	10	20	50	100	200
System users in total	10	100	500	1000	2000
OAuth 2.0 clients	3	10	25	50	100
Historical authentication requests	3000	30,000	150,000	300,000	600,000

The experiment was carried out using the AWS cloud computing resources and services. The following configuration of the infrastructure was created for the experiment purposes:

- one db.t3.medium type PostgreSQL 12.2-R1 RDS server;
- one t2.medium (2 vCPU, 2GB RAM) type EC2 server for data generation API application;
- one t2.medium (2 vCPU, 4GB RAM) type EC2 server for authentication API application.

The experiment was performed as follows:

- an infrastructure of a specified configuration was created on the AWS cloud computing platform;
- an authentication API application was installed into EC2 server and runs there;
- a database schema was created on the RDS server;
- a data generation API application was installed on the EC2 server;
- the configuration of the generated data set was specified and the data generation application was launched;
- conviction that the data set was generated successfully was done by connecting to the RDS;
- JMeter tool was installed and configured on the EC2 server, where a data generation API application was installed;
- JMeter tool was activated with the specified dataset configuration;
- the results of the experiment were collected from the *reporting/index.html* file upon a successful completion of the tests.

The experimental process was repeated five times with a different configuration of data sets.

4.3. Results

The 90th percentile of response times for the API endpoints (see Table 1) in a case of tested different data sets are shown in Figure 4.

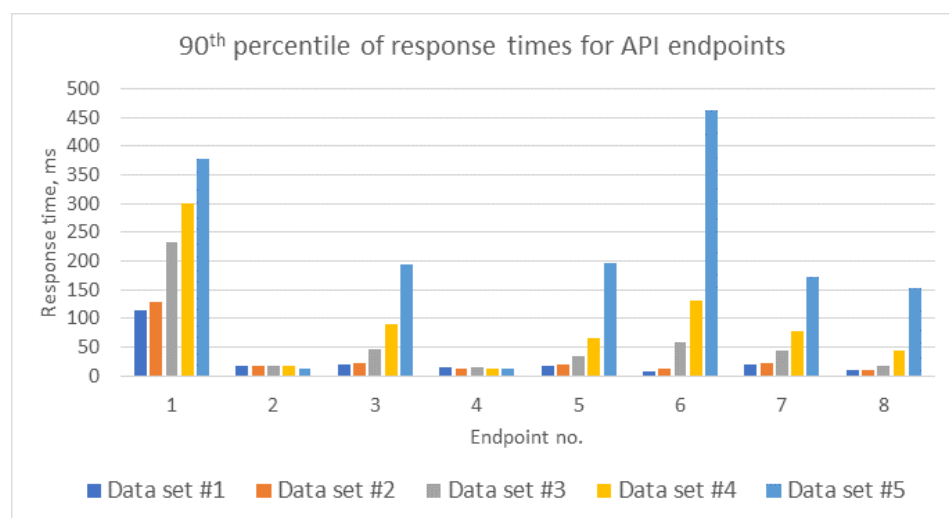


Figure 4. The 90th percentile of response times for the API endpoints.

To summarize, the following findings, based on the obtained experimental results, were determined:

- the number of database connections did not change during the experimental testing and it was 20. This number was constant, because the authentication API application is configured to use a fixed connection pool of a database;
- rising volumes of data sets affected the load on CPU of EC2 server—it also increased (i.e., the load of database reached a peak of 70.4% while the load rised up to 66.2% on EC2 server when data set #5 was tested);
- the maximum values in response times were carried out during the experiments with the first API endpoint (Create authentication request);
- the most noticeable changes in response times were between data sets #4 and #5, while other response times increased several times for most access points;
- testing with data set #4, which was configured for 100 parallel users, the response times for all API endpoints did not exceed the threshold of 300 ms. The results under

this perspective confirmed that the rapidity of the proposed method follows arised requirement for the operation of the system with 100 parallel users;

- testing with the largest data set #5, which was configured for 200 parallel users, the API endpoint with the highest response time processed requests in 462.9 ms. In this case, the same requirement for the operation of the system was met.

5. Experimental Testing of Resilience to Cyber Threats

Next, 17 penetration testing scenarios were developed in order to evaluate if the 2FA method-driven critical system is resilient to certain vectors of cyber-attacks.

5.1. Testing Setup

In order to automate penetration testing, a separate application based on a Spring Boot framework in Java programming language was written. The application for penetration testing provided the following functionality:

- entering the data required for the scripts into a database;
- accessing authentication API endpoints with HTTP requests that are specified in scripts;
- processing of testing scenarios and presentation of testing results.

The developed penetration testing scenarios allowed us to assess whether the predicted cyber-attacks could cause security vulnerabilities in 2FA method-driven critical system.

5.2. Testing Scenario

Firstly, the vectors of the cyber-attacks directly targeted to the main authentication API endpoints:

- `/api/oauth/request`—API endpoint for the initiation of the authentication request;
- `/api/oauth/request/{requestId}/approve-authentication`—API endpoint for the authentication approve;
- `/api/oauth/request/{requestId}/approve-authorization`—API endpoint for the authorization approve;
- `/api/oauth/token`—API endpoint for token issuance.

The vectors of cyber-attacks targeted at API endpoint for the initiation of the authentication request are described in Table 3.

Nine test scenarios were applied for API endpoints for the authentication as well authorization approve. In these scenarios, cyber-attacks acted as follows:

1. Scenario *Invalid_Signature*—Malevolent users could tamper JWT token payload, which would result in an invalid signature part;
2. Scenario *None_Algorithm*—Malevolent users could tamper JWT token by changing its signing algorithm to “None” to bypass signature verification;
3. Scenario *Tamper_Not_Before*—Malevolent users could tamper JWT token by changing its issuing time claim to use it before it becomes valid;
4. Scenario *Authorization Approve Other_Request*—Malevolent users could tamper JWT token by changing the user identifier stored within it to approve requests for other users that were not their subordinates;
5. Scenario *Authentication Approve Other_Request*—Malevolent users could tamper JWT token by changing the user identifier stored within it to approve the request on behalf of the other user;
6. Scenario *Tamper_Expiration*—Malevolent users could try to use a JWT token that had expired;
7. Scenario *Tamper_Action*—Malevolent users could tamper JWT token by changing its action claim to perform an action that they were not authorized to (i.e., to approve authentication);
8. Scenario *Replay*—Malevolent users could perform replay attacks to take advantage of an intercepted valid request;

9. Scenario *Expired_Session*—Malevolent users could submit a valid request after the authentication process session expiration time.

Table 3. Scenarios of cyber-attacks to */api/oauth/request* endpoint.

Scenario No.	Description of HTTP Requests (Request Method—POST)
<p>Scenario no. 1: Brute Force Password Malevolent users can use a brute-force attack to guess user passwords</p>	<p>Body:</p> <pre>{ "password": "password", "clientId": "security-test-client-3", "redirectUrl": "test-redirect-url", "codeChallenge": "ba7816bf8f01cfea414140de5dae2223b 00361a396177a9cb410ff61f20015ad", "username": "test-user" }</pre>
<p>Scenario no. 2: Tamper Redirect Url Malevolent users can tamper the „<i>redirectUrl</i>“ parameter and redirect users to malicious websites after successful authentication</p>	<p>Body:</p> <pre>{ "password": "password", "clientId": "security-test-client-3", "redirectUrl": "tampered-redirect-url", -- Tampered value "codeChallenge": "ba7816bf8f01cfea414140de5dae 2223b00361a396177a9cb410ff61f20015ad", "username": "test-user" }</pre>
<p>Scenario no. 3: Tamper Client ID Malevolent users can tamper the „<i>clientId</i>“ parameter to gain access to the system which should not be accessed by the user</p>	<p>Body:</p> <pre>{ "clientId": "security-test-client-4", -- Tampered value "code": "433182cc-0c7f-4cf0-bdf9- 2452a5d47d5e", "codeVerifier": "abc", "clientSecret": "security-test-client-secret", "grantType": "authorization_code" }</pre>

Token issuance is one of the key processes in proposed 2FA authentication method. For this reason, five testing scenarios with different vectors of the cyber-attacks were related with */api/oauth/token* endpoint (see Table 4).

Table 4. Scenarios of cyber-attacks to `/api/oauth/token` endpoint.

Scenario No.	Description of HTTP Requests (Request Method—POST)
<p>Scenario no. 1: Expired Refresh Token Malevolent users can use an expired refresh token to get a new access token and gain a never-ending session</p>	<p>Body:</p> <pre>{ "clientId": "security-test-client-3", "clientSecret": "security-test-client-secret", "grantType": "refresh_token", "refreshToken": "bb4f451e-a576-4312-97e8-179821a2c2e1" -- Tampered value }</pre>
<p>Scenario no. 2: Tamper Client ID Malevolent users can tamper the <code>clientId</code> parameter to get access tokens to the system which should not be accessed by the user</p>	<p>Body:</p> <pre>{ "clientId": "security-test-client-4", -- Tampered value "code": "a2c9959b-a741-4bb4-a907-55301909146e", "codeVerifier": "abc", "clientSecret": "security-test-client-secret", "grantType": "authorization_code" }</pre>
<p>Scenario no. 3: Tamper Client Secret Malevolent users can tamper the <code>clientSecret</code> parameter to gain access to the system which should not be accessed by the user</p>	<p>Body:</p> <pre>{ "clientId": "security-test-client-3", "code": "8536a120-2ab5-4bab-af0c-fed009b1ba0e", "codeVerifier": "abc", "clientSecret": "invalid-secret", -- Tampered value "grantType": "authorization_code" }</pre>
<p>Scenario no. 4: Tamper Code Verifier Malevolent users can tamper the <code>codeVerifier</code> parameter to bypass the <i>Proof Key for Code Exchange</i> verification check</p>	<p>Body:</p> <pre>{ "clientId": "security-test-client-3", "code": "ee3fec73-bcb3-4dcf-a79e-8a4f8fdd53ff", "codeVerifier": "tampered-verifier", -- Tampered value "clientSecret": "security-test-client-secret", "grantType": "authorization_code" }</pre>
<p>Scenario no. 5: Tamper Replay Malevolent users can perform replay attacks to take advantage of an intercepted valid request</p>	<p>Body:</p> <pre>{ "clientId": "security-test-client-3", "code": "610aaddb-b049-4e4e-bfae-00b9eee8d255", "codeVerifier": "abc", "clientSecret": "security-test-client-secret", "grantType": "authorization_code" }</pre>

5.3. Results

In general, the testing results of the resilience of 2FA method-driven critical system to cyber-attacks showed, that the system was protected from the vector of an identified attacks.

During the experimental penetration testing for API endpoint for the initiation of the authentication request, a brute-force attack was imitated by making three requests with an invalid password and one request was made with a valid password. The system responded

with an error, indicating that the account has been locked. Such a result was expected as after three unsuccessful authentication attempts the account of the user was locked and could be unlocked only by the administrator. Moreover, the system responded with an error in a case of tampered “*redirectUrl*” parameter, indicating that the provided “*redirectUrl*” parameter is invalid. The same happened with the tampered “*clientId*” parameter resulting system’s error, indicating that the user could not gain remote access to the critical system.

The next important findings showed that it was not possible access remotely to the critical system with maliciously forged token signature, a token signing algorithm, a token issue time claim, a tampered token or an expired token. In these cases all responses from the system were with errors, indicating that the token could not be processed successfully.

Going further, two forged requests were made during the experimental testing on scenario no. 7 applied for API endpoints for the authentication as well authorization approve. In the first one supervisor attempted to approve the authentication of another user and in the second one user tried to approve his authorization. The system responded with errors, indicating that the provided requests could not be processed successfully. The same approval request was made two times to imitate the replay attack. For the second request, the system responded with the similar error, as in previous tests. Besides these, an authentication session with an expired session time was created during the testing scenario no. 9. After receiving the request, the system responded with an error, indicating that the token could not be processed successfully because the session had expired.

It is important to note that results from testing scenario no. 8 were unexpected. A forged request with the tampered token owner was made during this test (see Table 5).

During the first test run, the system successfully accepted forged token and approved authentication. After that, the identified security issue was fixed (Table 6).

Another findings were carried out after the penetration testing related with API endpoint */api/oauth/token*. After receiving the request with an expired refresh token, the system responded with an error, indicating that the request could not be processed successfully, because the provided refresh token was invalid. When a forged request with the tampered “*clientSecret*” parameter was made, the system also responded with an error, indicating that the provided OAuth 2.0 client was not found. The similar response was provided in a case of the tampered “*codeVerifier*” parameter during the authentication process. The same token exchange request was made two times to imitate the replay attack during the test. All the times the system responded with an error, indicating that the associated authorization code was not found (i.e., it had already been used).

Table 5. HTTP request during testing scenario no. 8.

Description	Body
The JWT token provided in the request is decoded	<pre>{ "kid": 3631, -- Tampered value "alg": "RS256" }</pre>
HTTP response body	<pre>{ "action": "APPROVE_AUTHENTICATION", "iat": 1614949467, "sub": "3631", "exp": 1614949647 } { "token": "eyJraWQiOiJ2MzEsImFsZyI6IjR2In0..." }</pre>

Table 6. HTTP responses during testing scenario no. 8.

Description	Body
False positive	<pre>HTTP/1.1 200 Content-Type: application/json { "status": "success", "data": { "requestStatus": "PENDING_AUTHORIZATION_APPROVAL", "requestId": "7FF10BCA" } }</pre>
After fixing	<pre>{ "status": "error", "data": { "code": "error.validation", "message": "Failed to verify token" } }</pre>

The next unexpected results appeared during the testing scenario no. 2 related with API endpoint for token issuance. A forged request with the tampered “clientId” parameter was made during this test (see Table 4). During the first test run, the system successfully accepted the forged request and issued an access token to the system which should not be accessed by the user (Table 7).

Table 7. HTTP responses during testing scenario no. 2.

Description	Body
False positive	<pre>HTTP/1.1 200 Content-Type: application/json { "status": "success", "data": { "accessToken": "eyJraWQiOiIiIiwiaWF0IjoiMTI1MTIifQ... ", "refreshToken": "17d0ab1c-a7c3-4309-a002-5abb3ebd0784" } }</pre>
After fixing	<pre>{ "status": "error", "data": { "code": "error.validation", "message": "Oauth client was not found" } }</pre>

The identified security issue was fixed.

6. Conclusions

In this paper, a 2FA authentication method has been proposed and evaluated, demonstrating significant benefits in managing remote access to an information system of the critical infrastructure. The testing results showed, that the proposed method can protect a critical system from dangerous HTTP requests and its publicly available API endpoints would be protected from threatening inputs that could cause malicious activities on the critical infrastructure. The rapidity of the method could ensure the stable operation of the system for certain number of users working in parallel.

Author Contributions: Conceptualization, K.J. and R.B.; methodology, R.B.; software, K.J.; validation, K.J., R.B.; formal analysis, R.B.; investigation, K.J.; resources, K.J.; writing—original draft preparation, K.J.; writing—review and editing, R.B.; visualization, K.J. and R.B.; supervision, R.B. Both authors have read and agreed to the published version of the manuscript.

Funding: Nordplus-Advances in Information, Automation and Electrical Engineering (ENERGY-COM). NPHE 2020/10059.

Data Availability Statement: The data presented in this paper are available on request from the corresponding author. The data are not publicly available due to the project is not completed.

Acknowledgments: The authors express special thanks to prof. Algimantas Venckauskas from Kaunas University of Technology for his support, recommendations and guidance during this work. Some ideas in this paper are part of the outcomes of Nordplus project NPHE 2020/10059-Advances in Information, Automation and Electrical Engineering.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- Mullane, M.A. Cyber Attacks Targeting Critical Infrastructure. Available online: <https://etech.iec.ch/issue/2019-02/cyber-attacks-targeting-critical-infrastructure> (accessed on 16 October 2019).
- Adelmeyer, M.; Teuteberg, F. Cloud Computing Adoption in Critical Infrastructures-Status Quo and Elements of a Research Agenda. In Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI 2018), Lüneburg, Germany, 6–9 March 2018; pp. 1345–1356.
- National Intelligence Strategy of the United States of America. Reports and Publications. 2019. Available online: <https://www.dni.gov/index.php/newsroom/reports-publications/item/1943-2019-national-intelligence-strategy> (accessed on 22 September 2020).
- Kaspersky Lab ICS CERT. Threat landscape for Industrial Automation Systems (Report H1 2020). Available online: <https://ics-cert.kaspersky.com/reports/2020/09/24/threat-landscape-for-industrial-automation-systems-h1-2020/> (accessed on 13 November 2020).
- Archana, B.S.; Chandrashekar, A.; Bangi, A.G.; Sanjana, B.M.; Akram, S. Survey on usable and secure two-factor authentication. In Proceedings of the 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT), Bangalore, India, 19–20 May 2017; pp. 842–846.
- Ometov, A.; Bezzateev, S.; Mäkitalo, N.; Andreev, S.; Mikkonen, T.; Koucheryavy, Y. Multi-factor authentication: A survey. *Cryptography* **2018**, *2*, 1. [[CrossRef](#)]
- Ali, F.A.B.H.; Hanza, M.Z.B.M.; Sukri, M.A.B.M. Two Factor Authentication by Using SMS for Web Based Application. *Int. J. Inf. Technol.* **2020**, *9*, 21–24.
- Drzhzhin, A. SMS-Based Two-Factor Authentication Is Not Safe—Consider These Alternative 2FA Methods Instead. Available online: <https://www.kaspersky.com/blog/2fa-practical-guide/24219/> (accessed on 17 January 2019).
- Grassi, P.A.; Fenton, J.L.; Burr, W.E. Digital Identity Guidelines—Authentication and Lifecycle Management: NIST Special Publication 800-63B. Available online: <https://pages.nist.gov/800-63-3/sp800-63b.html> (accessed on 10 January 2019).
- Markert, P.; Farke, F.; Dürmuth, M. View the email to get hacked: Attacking SMS-based two-factor authentication. In Proceedings of the WAY Conference, Santa Clara, CA, USA, 11 August 2019; pp. 1–6.
- Babkin, S.; Epishkina, A. Authentication protocols based on one-time passwords. In Proceedings of the 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EConRus), Saint Petersburg and Moscow, Russia, 28–31 January 2019; pp. 1794–1798.
- Pernpruner, M.; Carbone, R.; Ranise, S.; Sciarretta, G. The Good, the Bad and the (Not So) Ugly of Out-of-Band Authentication with eID Cards and Push Notifications: Design, Formal and Risk Analysis. In Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, 16–18 March 2020; pp. 223–234.
- Bissada, A.; Olmsted, A. Mobile multi-factor authentication. In Proceedings of the 12th IEEE International Conference for Internet Technology and Secured Transactions (ICITST), Cambridge, UK, 11–14 December 2017; pp. 210–211.
- Aldumijji, N.A.; Khan, E.A. Fingerprint and location based multifactor authentication for mobile applications. *Int. J. Eng. Technol.* **2019**, *8*, 193–204.
- Zhang, F.; Kondoro, A.; Muftic, S. Location-based authentication and authorization using smart phones. In Proceedings of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, UK, 25–27 June 2012; pp. 1285–1292.
- Bhand, A.; Desale, V.; Shirke, S.; Shirke, S.P. Enhancement of password authentication system using graphical images. In Proceedings of the IEEE International Conference on Information Processing (ICIP), Pune, India, 16–19 December 2015; pp. 217–219.

17. Das, S.; Dingman, A.; Camp, L.J. Why Johnny doesn't use two factor a two-phase usability study of the FIDO U2F security key. In *Financial Cryptography and Data Security*; Meiklejohn, S., Sako, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 160–179.
18. Choi, Y.; Lee, Y.; Moon, J.; Won, D. Security enhanced multi-factor biometric authentication scheme using bio-hash function. *PLoS ONE* **2017**, *12*, 1. [[CrossRef](#)] [[PubMed](#)]
19. Mahadi, N.A.; Mohamed, M.A.; Mohamad, A.I.; Makhtar, M.; Kadir, M.F.A.; Mamat, M. A survey of machine learning techniques for behavioral-based biometric user authentication. In *Recent Advances in Cryptography and Network Security*; Mitra, P., Ed.; IntechOpen: London, UK, 2018; pp. 43–54
20. Corradini, F.; Ferrari, A.; Fornari, F.; Gnesi, S.; Polini, A.; Re, B.; Spagnolo, G.O. A guidelines framework for understandable BPMN models. *Data Knowl. Eng.* **2018**, *113*, 129–154. [[CrossRef](#)]
21. Bonneau, J.; Herley, C.; Van Oorschot, P.C.; Stajano, F. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–23 May 2012; pp. 553–567.
22. Jurgilas, K. Subjekto 2FA skaitmeninio autentifikavimo prie kritinės infrastruktūros informacinės sistemos struktūrizuotas vertinimas. In Proceedings of the conference “Lietuvos magistrantų informatikos ir IT tyrimai”, Vilnius, Lietuva, 14 May 2021; pp. 23–33.
23. Boonkrong, S. Internet Banking Login with Multi-Factor Authentication. *KSII Trans. Internet Inf. Syst.* **2017**, *11*, 511–535.
24. Hussein, K.W.; Sani, N.F.M.; Mahmud, R.; Abdullah, M.T. Design and Implementation of Multi Factor Mechanism for Secure Authentication System. *Int. J. Comput. Sci. Inf. Secur.* **2013**, *11*, 31–37.
25. Lami, I.A.; Kuseler, T.; Al-Assam, H.; Jassim, S. LocBiometrics: Mobile phone based multi-factor biometric authentication with time and location assurance. In Proceedings of the Telecommunications forum TELFOR, Serbia, Belgrade, 23–25 November 2010; pp. 151–154.
26. Maciej, B.; Imed, E.F.; Kurkowski, M. Multifactor Authentication Protocol in a Mobile Environment. *IEEE Access* **2019**, *7*, 157185–157199. [[CrossRef](#)]
27. Abdellaoui, A.; Khamlichi, Y.I.; Chaoui, Y. A Novel Strong Password Generator for Improving Cloud Authentication. *Procedia Comput. Sci.* **2016**, *85*, 293–300. [[CrossRef](#)]
28. Fang, X.; Zhan, J. Online Banking Authentication Using Mobile Phones. In Proceedings of the 5th International Conference on Future Information Technology, Busan, Korea, 21–23 May 2010; pp. 1–5.
29. Misbahuddin, M.; Roshni, V.; Thomas, A.; Kumar, U. A Unique-ID based Usable Multi-Factor Authentication Scheme for e-Services. In Proceedings of the International Conference for Security and Management, Las Vegas, NV, USA, 27–30 July 2015; pp. 295–301.