IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 10, 2022, accepted January 24, 2022, date of publication February 1, 2022, date of current version February 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3148537

User Authentication Protocol Based on the Location Factor for a Mobile Environment

MACIEJ BARTŁOMIEJCZYK[1], IMED EL FRAY[1], MIROSŁAW KURKOWSKI[2], SABINA SZYMONIAK[3], AND OLGA SIEDLECKA-LAMCH[3]

[1] West Pomeranian University of Technology, 70-310 Szczecin, Poland
[2] Institute of Computer Science, Cardinal Stefan Wyszynski University in Warsaw, 01-815 Warsaw, Poland
[3] Department of Computer Science, Czestochowa University of Technology, 42-201 Czestochowa, Poland

Corresponding author: Imed El Fray (ielfray@zut.edu.pl)

ABSTRACT The way the internet is used by billions of users around the world has been revolutionized by mobile devices. The capabilities of smartphones are constantly growing, and the number of services available for mobile devices is also increasing. This undeniable trend makes smartphones terminals for accessing services that process confidential data, which make smartphones priceless targets of cyberattacks. Along with an increasing number of mobile services, the methods of securing the confidentiality, integrity and availability of systems used have also evolved and adapted to the capabilities of a mobile environment. One of the important security services is the user authentication process. This process often implements the postulates of strong authentication, multistage authentication based on factors from the knowledge, position and inherence categories. Unfortunately, the implementation of the factors belonging to these categories is not always possible due to the limitations of smartphones, such as the lack of interfaces for the implementation of biometrics or environmental factors - problems with network or internet access in various countries and regions. Therefore, there is a need to analyse the possibility of implementing a strong authentication process based on additional information about users, e.g., based on location data. The article analyses the requirements for the authentication process and authentication factors. Based on the performed analysis, the criteria that each authentication factor must meet were defined. This article presents a proposal for a user authentication protocol based on the location factor for a mobile environment. The method can be used in the case of problems with the implementation of strong authentication or as an additional authentication factor that increases the security of the user identity confirmation process. The presented protocol has been analysed in terms of performance, security and compliance with the requirements related to the authentication factors.

INDEX TERMS Authentication protocols, electronic identification, mobile environment, multifactor authentication, location-based authentication.

I. INTRODUCTION

The development of mobile technology has made smartphones terminals that allow the implementation of many key services, such as access to confidential information. For this reason, the user authentication process must be performed as securely as possible. Secure authentication of user identities is performed using multifactor authentication, a process that requires the use of more than one factor. This process can be conducted based on a user's knowledge, something that he possesses or a biometric feature [1]. The definition of the strong authentication process is very similar. It consists of confirming a user's identity based on at least two factors belonging to different categories: knowledge, possession, and inherence [2], [3]. The abovementioned definitions show that the key to a secure authentication process is to ensure the greatest number of factors from different categories. Strong authentication is usually conducted by forcing the user of, e.g., e-banking, e-administration, etc. to enter a password (knowledge) and confirm possessing a device assigned to the user to which a one-time code is sent (possession). Undoubtedly, the advantages of this method include the simplicity of the implementation of the authentication mechanism [4]–[6]. Many systems are relying on this factor to

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masini.

comply with SCA requirements [2], [7]. However, there are many sources indicating that attacks on OTPs (One Time Passwords), such as SIM cloning, SIM swapping, and message intercepting, are real threats to the identity confirmation process [8], [9]. Moreover, if a one-time code is sent to a user as an SMS message, the range of the cellular network is required to perform the authentication process. If the code is delivered in the form of a push message, the user needs access to the internet. Therefore, the lack of network access may make strong user authentication impossible. Alternatively, the strong authentication process in a mobile environment uses a biometric element based on a fingerprint or facial scan of a user. The use of biometrics in the process of confirming a user's identity is the subject of numerous studies [10]–[13]. This method requires a device with a specific reader. Despite its popularity, a large group of users have devices that do not support biometric authentication. Therefore, to implement multifactor authentication in the absence of internet access or cellular network coverage, the factor of another group should be used.

As part of the research conducted by the authors of this article, which is a continuation of research on the implementation of security services in a mobile environment [14], numerous studies on the authentication process and authentication factors were analysed. Based on the analysed sources, the authors defined the requirements that must be met by an authentication factor. Based on the presented requirements, a new group of authentication factors based on a user's location data has been presented. There are many studies that seek to define the authentication process. Each of them indicates that the authentication process consists of confirming a user's identity [1], [2], [3], [15]. Such a definition of authentication can be used to define one of the requirements related to an authentication factor. An authentication factor must uniquely identify the user. Further requirements result from the definition related to the authenticator. An authenticator is defined as a factor that a user possesses or controls and can be used to authenticate a subject's identity [1]. The definition emphasizes the need to associate an authentication factor with a user. Based on this definition, a further requirement for an authentication factor can be defined. An authentication factor must be controlled by a user. Moreover, an authentication factor must be bound to exactly one user. According to the definitions included in the presented sources, authentication is the process of verifying the identity declared by a user. During this process, the pattern data provided to the system during the registration process are compared with the data provided during the authentication attempt to confirm that the user's identity is true. The identity confirmation process can be conducted based on a factor from the categories of knowledge, possession or biometrics. The analysed regulations define the requirements for the enrolment phase [1], [3]. These requirements indicate that the system performing the user authentication process must collect the appropriate data required to confirm and verify a user's identity. Therefore, it should be ensured that the pattern data that will be compared during the identity confirmation process (Table 1) will be obtained and associated with a user before the authentication process is conducted.

TABLE 1. Authentication factor requirements.

REQUIREMENT	Knowledge factor	Possesion factor	Inherence factor
Verify user identity	*yes*	*yes*	*yes*
Controlled by the user	*yes*	*yes*	*yes*
Bound to the user	*yes*	*yes*	*yes*
Comparable with the reference pattern data	*yes*	*yes*	*yes*

The conducted analysis shows that the authentication factors from the knowledge, possession and inherence categories meet the criteria established based on the quoted norms and standards. Therefore, if the proposed solution based on a user's location data is to be an authentication factor or a supporting authentication factor, it must also meet the criteria summarized in Table 1. The collected requirements can help to assess whether the proposed authentication factor using a user's location data can be treated as an authentication factor and whether it truly performs the confirmation process of a user's identity and not the task of authorizing access to services. Solutions that use information about a user's location are gaining increasingly more popularity. The use of a location interface is associated with numerous problems, such as the problem with determining the identity of an authenticated entity or relatively low accuracy of information about a location. Moreover, information about a user's location is susceptible to spoofing at the hardware level or at the operating system level [16]. Solving the problem of obtaining a confirmed device location is not the same as solving the issue of using location as an authentication factor as long as the location is not bound to user identity. The use of coordinates representing a user's current location requires in-depth research into how this information can be used to confirm a user's declared identity.

II. RELATED WORKS

Many publications discuss the use of location data to increase the level of security. One way to use location data is to use them as a knowledge factor. A location is information remembered by a user and is entered as a type of password [17]. Using location information in such a way is an authentication process but must not be considered authentication based on location data. In this case, the location plays the role of the knowledge factor. The availability of a GPS interface and the ease of its use in determining the exact location increases the use of location data in processes conducted in a mobile environment. When analysing the latest publications, one of the directions of the user identity confirmation process is online authentication, also called continuous authentication [18]. Online authentication uses data collected by your smartphone and may be based on your location data. Data from a GPS interface or the location determined based on the Wi-Fi network are used by properly prepared classifiers. Based on these data, the algorithm determines whether the person who owns the device is its owner [19], [20]. Not all analysed publications use location data to perform authentication. Granting access to resources when staying in a declared city is the implementation of the authorization service and not confirming a user's identity. Some of the analysed protocols address the topic of confirming the accuracy of location data. The purpose of such solutions is to confirm that the location data are correct and unmodified. However, such information is insufficient to determine a user's identity. One of the ways to obtain information about the location of a device associated with a user is to determine its location based on the delay of the signal sent from the device to the surrounding network devices, e.g., Wi-Fi routers [21]. A user's location can be determined based on the IP address of the router to which a smartphone is connected. The location data obtained from the network provider are used to grant access to the service if the user is in the area declared during user account registration [22]. The analysed protocols actually use data on a user's location, but granting access based on the current location is the implementation of the authorization service, not authentication. Another publication [23] uses location data for the authentication process. However, these coordinates are not associated with a specific user but constitute additional information that can be used by a server to authorize access. As in the case of the previous article, another source [24] presents a scheme based on location data. It makes decisions about granting access to a service. Authorization is performed depending on whether the sent location is correct and whether there is any doubt about its modification. It should be emphasized that the published research lacks proposals regarding the use of user location data for user authentication. That is why this issue is taken up in our article.

In addition, it should be noted that the use of location data may pose a security risk. Information about a user's location can be very valuable to attackers. These issues have led many researchers to address the issue of using location data and ensuring their confidentiality at the same time [23]–[25]. The protocols proposed in these publications use location data and show how to use location data in a way that guarantees their safety in the event of a leak. One of the ways to ensure the confidentiality of a location is to use a schema that generates a k-1 "dummy location" [25]. In such a situation, the probability of information leakage about a user's location is reduced to 1/k. The confidentiality of user location data can be ensured by using encryption [23]. In the presented scheme, location data are sent to the server after they have been encrypted with a shared key. Another of the analysed solutions ensures the confidentiality of location data by encrypting the transmitted information [24]. The difference lies in the method of generating the key, which is established between the mobile application and the server. The key is generated based on the carrier frequency offset (CFO) resulting from the use of the Wi-Fi network. The obtained key is used to encrypt the frame consisting of the preamble, MAC header and transmitted data. The issue of using a user's location data with simultaneous location privacy has been considered in our research. Our protocol was designed to meet the requirements of location privacy. The presented solutions [23]–[25] increase the level of security of services but are limited to managing access to services based on data on a user's current location. Moreover, solutions that use only a location determined based on a Wi-Fi network face the problem of location accuracy. This problem prevents the use of location in the authentication process [24]. That is why our solution uses information about the location of the device based on three independent sources: GPS, Wi-Fi and data from the GSM network.

The computing power of smartphones has increased significantly, and these devices are equipped with an increasing number of various types of hardware interfaces. These conditions contributed to numerous studies on new, dedicated security mechanisms in a mobile environment. One of the interfaces from which data can be used to increase the security level is the GPS module. Due to the popularity of the use of location data positioning mechanisms, they have gained increasingly more importance. As research shows, efficiently determining the locations of devices within the assumed radius is not a difficult task. Location accuracy is influenced by the development of new technologies, including the growing popularity of 5G networks. An analysed publication [26] shows that the use of next-generation networks may increase the location accuracy based on the data of the cellular network. The proposed algorithm allows the exact location of a device inside a building to be determined. The location of a device is determined by dividing the coverage area into sectors using a UPA (uniform planar array). Data obtained from GPS, cameras, fingerprint readers, and the NFC interface can be used for the user authentication process. NFC enables communication between two devices, ensuring that these devices are in close proximity to the terminal [27]. One of the analysed multifactor authentication protocols [28] uses the knowledge factor, biometrics and the NFC interface as an implementation of the possession factor. NFC is used for communication between the devices of the sender and the payee. This protocol requires the recipient to confirm the identity of the sender of a payment.

According to the authors, by using the computing power of smartphones and their numerous interfaces (GPS and NFC), properly used data of a user's current location may constitute the fourth authentication factor that does not belong to any of the commonly known groups of authentication factors based on knowledge, possession, inherence. A similar approach was proposed in [29]. The authors of the article propose implementing a process that can be called authentication, not just authorization. The protocol uses location data obtained from a user's device and confirmation of its location by other users acting as witnesses. Packets prepared in this way are sent to an authentication server where the provided data are verified. It should be noted that this scheme authenticates not the user but the location data. This means that the location data, not the user's identity, are confirmed.

III. THE PROPOSED MULTIFACTOR AUTHENTICATION SCHEME

Considering the analysed publications, other research on multifactor authentication methods and the possibility of using location data, we present an authentication scheme based on information about a user's location. The protocol implements the identity confirmation process in two phases. The first stage consists of dynamically declaring a trusted area for a given user u_i and temporarily assigning this area to only one user. The area is assigned to the user until authentication is completed or the EXP time (see Table 2) expires. The second step confirms that the authenticated user is in the predefined area. Information about the location of the authenticated user is obtained from various sources. A user's location data are confirmed by witness applications installed on the devices of other users nearby. During the first stage, called the declaration of a trusted area, the location is determined based on the interfaces of the mobile device (GPS, Wi-Fi, and cellular network). The coordinates used in the stage of confirming the declared location are obtained from a trusted source against which the user authenticates. For the purposes of the protocol, we assumed that this trusted source may be the POS (point of sale) terminal where the payment is made. The proposed authentication scheme is an extension of the multifactor authentication protocol described in the article "Multifactor Authentication Protocol in a Mobile Environment" [14]. The main goal of the research is to increase the level of security of the user identity confirmation process. One way to do this is to increase the number of authentication factors. The level of security can be increased by using factors from different groups (knowledge, possession, and inherence). The proposed protocol shows only the part of the authentication process that uses location data. It should be emphasized that the proposed protocol complements the previously presented multifactor authentication protocol in a mobile environment and can be implemented in the next stage of authentication. Alternatively, the proposed protocol can be used as an authentication factor from a different group in the case of limitations that prevent the use of other authentication factors, e.g., lack of biometric interfaces (inherence factor), no GSM network coverage to receive an SMS (possession factor) or no access to the internet (sending a push message).

A. SYSTEM ARCHITECTURE

A secure authentication protocol should be designed considering the possibilities of the environment in which it will be implemented. In the case of a mobile environment, security mechanisms provided by mobile platforms, e.g., Android, are extremely helpful. The presented authentication protocol based on location data is closely related to the capabilities of the mobile platform. The presented solution consists of the mobile application (MA) of user u_i, which is authenticated; and the witness application (WA) installed on the device of another user located near user u_i. The authentication process also uses a POS (POS) terminal, which is located in the place where the user attempts to authenticate. The last element of the system is a backend service consisting of an authentication service (AS), which uses firebase cloud messaging (FCM) as a notification service [30]. All elements of the system are shown in the diagram in Figure 1.

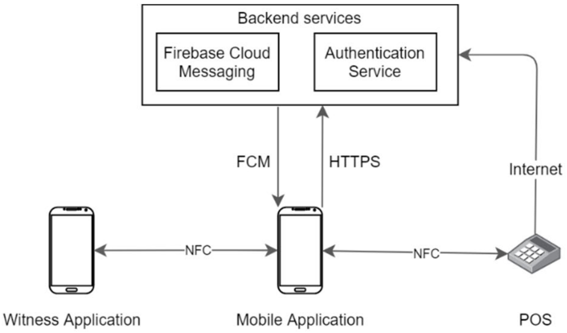


FIGURE 1. Diagram of communication between actors of the protocol.

B. PREREQUISITES

The presented protocol uses the assumptions and initial conditions necessary for its implementation. The requirements assume that the identity of the user being authenticated is known and that all necessary data have been obtained during the protocol initialization and user registration phases. "Necessary data" refers to the data provided upon a user registration, examples could be user identity, password hash, mobile application identifier and device identifier. The registration process and the phase of key sharing are out of the scope of the article's research

1) AUTHENTICATION SERVICES

1) AS stores data of registered user u_i who is identified by id_i.
2) AS store fcmToki associated with ui. fcmToki is identifier in the FCM service.
3) AS securely stores the private key KPR_AS used to decrypt.
4) AS known KP_MAE, KP_MAS, KP_WA and KP_POS can be used to encrypt and verify the signatures of messages.
5) AS knows the exact location of the POS and its identity.
6) If the first phase of the protocol is successful, the location sent by the user of the mobile application is locked. After the specified time has elapsed or the authentication process has finished, the location is available again to another user.

2) MOBILE APPLICATION

1) In the registration process, the application generates and stores two pairs of the RSA key with a length of 4096 bytes in the Android secure keystore. The first pair is used to implement digital signatures, and the second

TABLE 2. Notations and parameters of the protocol.

Symbol	Description
AS	Authentication service
MA	Mobile application installed on the device of user u$_i$ who is being authenticated
FCM	Firebase cloud messaging – push notification server
POS	POS terminal or device that is located at the authentication process place. The coordinates of the POS terminal are trusted and known. The POS has an interface supporting RFID communication.
WA	Android application launched in background on the mobile device of witness user who can confirm the location of user u$_i$.
SUUID	Session identifier using the UUID format [31]
SUUID$_G$	Subsession identifier using the UUID format [31] corresponding to the authentication process based on user location.
Enc$_k$(•)	Asymmetric key encryption with key k
Dec$_k$(•)	Asymmetric key decryption with key k
Sign$_k$(•)	Digital signature process with key k
Verify$_k$(•)	Digital signature verification process with key k
id$_i$	Identity of user u$_i$
KP_AS	Asymmetric public key corresponding to KPR_AS stored in AS and bound to AS. Used to encrypt messages sent to Authentication service.
KPR_AS	Asymmetric private key corresponding to KP_AS used to decrypt received messages..
KP_MAS	Asymmetric public key corresponding to KPR _MAS stored in AS and bound to MA. Used to verify signatures.
KPR_MAS	Asymmetric private key corresponding to KP_MAS generated and store in the Android keystore. Used to sign messages.
KP_MAE	Asymmetric public key corresponding to KPR_MAE stored in AS and bound to MA used to encrypt messages.
KPR_MAE	Asymmetric private key corresponding to KP_MAE generated and store in the Android keystore. Used to decrypt received messages.
KP_WA	Asymmetric public key corresponding to KPR _WA stored in AS and bound to MA. Used to verify signatures.
KPR_WA	Asymmetric private key corresponding to KP_WA generated and stored in the Android keystore. Used to sign messages.
KP_POS	Asymmetric public key corresponding to KPR_ POS stored in AS and bound to the POS. Used to verify signatures.
KPR_POS	Asymmetric private key corresponding to KP_ POS. Used to sign messages.
fcmTok$_i$	Firebase ID token bound to u$_i$ enable to send notifications with FCM
t$_s$	Current timestamp
t$_{sg}$	Object generation timestamp
t$_{sv}$	Object validation timestamp
EXP	Constant that defines the validity of the authentication factor

pair is used to decrypt the messages sent to the mobile application.

2) Because the proposed protocol consists of two phases, the application possesses two pairs of cryptographic keys. Diversification of the keys used may increase the security level of the protocol.
3) The application is able to use KP_AS to encrypt confidential data.
4) The application is installed on a device equipped with GPS. During the authentication process, the device is able to retrieve location data. The accuracy of the location data is very high.
5) The application is installed on a device equipped with an NFC interface. The user of the mobile application is able to place his own device close to the device of the user of the witness application and to the POS terminal.

3) WITNESS APPLICATION

1) In the registration process, the witness application generates and stores RSA keys with a length of 4096 bytes in the Android secure keystore. The private key is used to sign messages sent to the mobile application.
2) The application is able to use KP_AS to encrypt confidential data.
3) The application is installed on a device equipped with GPS. While supporting the authentication process of the user of the mobile application, the device is able to retrieve location data. The accuracy of the location data is very high.
4) The application is installed on a device equipped with an NFC interface. The user of the witness application is able to place his own device close to the device of the user of the mobile application.

C. PROPOSED USER AUTHENTICATION PROTOCOL

1) PHASE 1: DECLARATION OF TRUSTED USER LOCATION

Figure 2 shows the first phase of the user authentication process based on a factor referring to a user's location. The first phase of the process considers the dynamic way in which the trusted location point is determined. The first phase of the protocol guarantees the confidentiality of sensitive data of all users involved in the exchange of messages. This is a reason why location data are encrypted with the public key of authentication services. All messages exchanged between the mobile application and witness application shall be transmitted via the NFC interface. The use of NFC limits the distance between the user performing the authentication process and the witness application of another registered user.

Users start the geoauthentication process by pushing the button in mobile applications. The MA seeks to obtain the most accurate location data using GPS, GSM or IP addresses.

$$Launch\ geoauthentication \tag{1}$$

$$loc_A = getLocation(GPS, GSM, IP) \tag{2}$$

The mobile application encrypts the message consisting of id$_1$, session identifier SUUID$_G$, location loc$_A$, timestamp t$_{s1}$ and pseudorandom value RNDM. The prepared message is

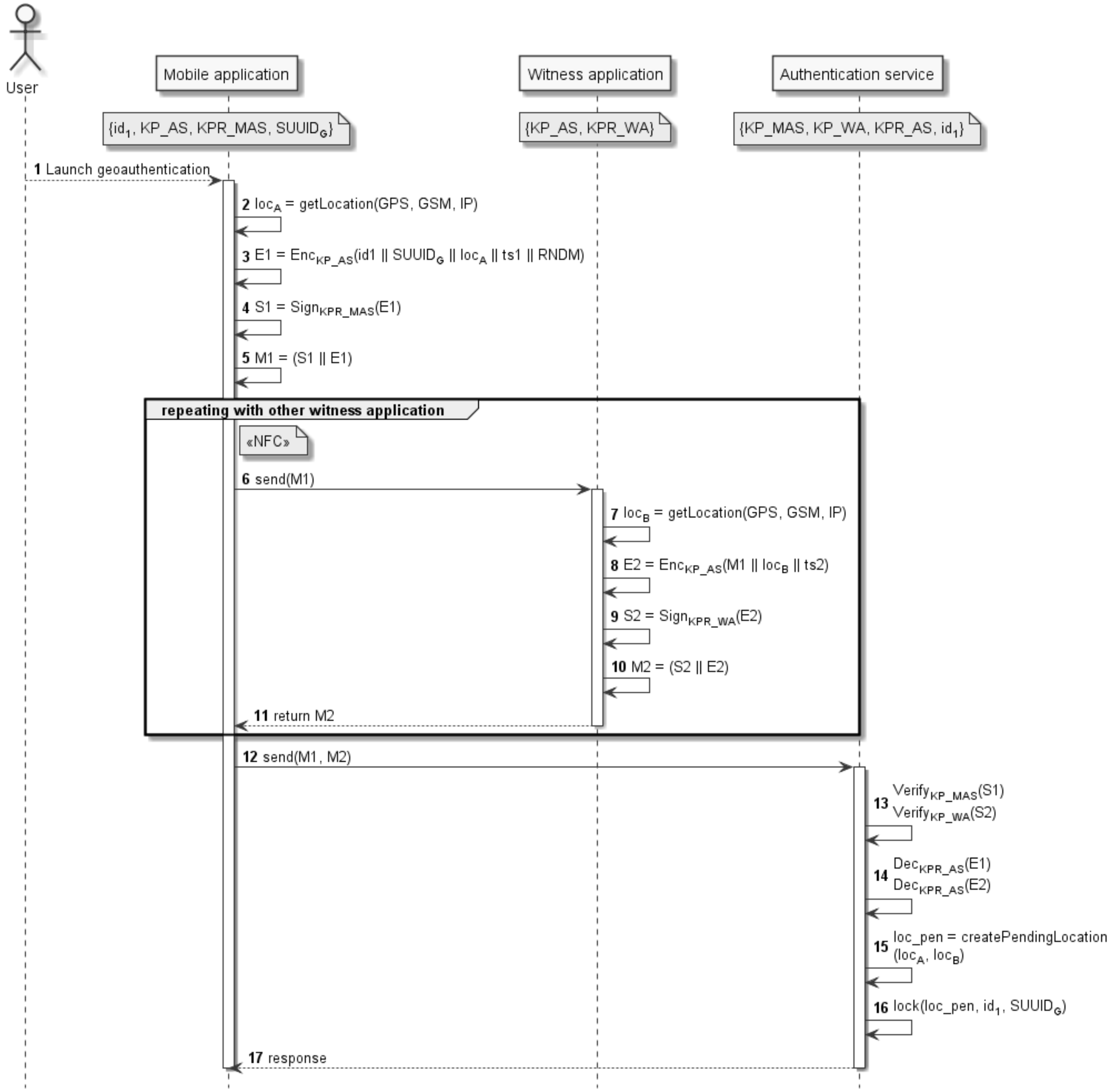

User
Mobile application
Witness application
Authentication service
{id_1, KP_AS, KPR_MAS, $SUUID_G$}
{KP_AS, KPR_WA}
{KP_MAS, KP_WA, KPR_AS, id_1}
1 Launch geoauthentication
2 loc_A = getLocation(GPS, GSM, IP)
3 E1 = Enc_{KP_AS}(id1 || $SUUID_G$ || loc_A || ts1 || RNDM)
4 S1 = $Sign_{KPR\_MAS}$(E1)
5 M1 = (S1 || E1)
repeating with other witness application
«NFC»
6 send(M1)
7 loc_B = getLocation(GPS, GSM, IP)
8 E2 = Enc_{KP_AS}(M1 || loc_B || ts2)
9 S2 = $Sign_{KPR\_WA}$(E2)
10 M2 = (S2 || E2)
11 return M2
12 send(M1, M2)
13 $Verify_{KP\_MAS}$(S1) $Verify_{KP\_WA}$(S2)
14 Dec_{KPR_AS}(E1) Dec_{KPR_AS}(E2)
15 loc_pen = createPendingLocation (loc_A, loc_B)
16 lock(loc_pen, id_1, $SUUID_G$)
17 response

FIGURE 2. Declaration of the user location.

encrypted with the authentication service's public key.

$$E_1 = Enc_{KP\_AS}(id_1||SUUID_G||loc_A||t_{s1}||RNDM) \tag{3}$$

The calculated value is signed with the mobile application's private key and sent to the witness application via NFC based on the HCE Android API [32]. M_1 is a message consisting of encrypted message M1 and signature S1.

$$M_1 = Sign_{KPR\_MAS}(E1) \tag{4}$$

$$M1 = (S1||E1) \tag{5}$$

$$send(M_1) \tag{6}$$

The witness application receives the message and seeks to obtain the most accurate location data loc_B.

$$loc_B = getLocation(GPS, GSM, IP) \tag{7}$$

The witness application encrypts the message consisting of received message M_1, location loc_B, and timestamp t_{s2}. The prepared message is encrypted with the authentication service public key.

$$E_2 = Enc_{KP\_AS}(M_1||loc_B||t_{s2}) \tag{8}$$

The calculated value is signed with the witness application's private key and returned to the mobile application.

The witness application prepares an M2 message consisting of encrypted value E2 and signature value S2.

$$S_2 = Sign_{KPR\_WA}(E2) \tag{9}$$

$$M2 = (S2||E2) \tag{10}$$

$$return(M_2) \tag{11}$$

The mobile application sends gained messages to the authentication service. The mobile application could send more than two messages received from the witness application instances. User location can be proven by more than one witness application.

$$send(M_1, M_2) \tag{12}$$

The authentication service decrypts and verifies the messages it has received. The first step of verification is to check the integrity of the processing data. Signature $S1_1$ was verified with KP_MAS, and signature S_2 was verified with KP_WA. AS decrypts message E_1 with the private key of the authentication service and message from the witness application instances E_2. Messages from other application are decrypted in the same way. AS extracts the location data of the mobile application and witness application instances and verifies them.

$$Verify_{KP\_MAS}(S1), \quad Verify_{KP\_WA}(S2) \tag{13}$$

$$Dec_{KPR\_AS}(E1), \quad Dec_{KPR\_AS}(E2) \tag{14}$$

Verified location data are used to set the pending location of user u_i. is the process proposes calculating user location based on all of the received coordinates. The pending location is the data that have to be confirmed in the second phase of the protocol.

$$loc_{pen} = createPendingLocation(loc_A, loc_B) \tag{15}$$

An authentication service verifies whether the location is not bound to another user. Then, AS locks the pending location for some time and bounds the location to identity id_1 and session $SUUID_G$. The location is blocked until the authentication process is completed.

$$lock(loc_{pen}, id_1, SUUID_G) \tag{16}$$

The authentication service returns to the mobile application status of the processed request.

$$Response \tag{17}$$

2) PHASE 2: CONFIRMATION OF DECLARED USER LOCATION

Figure 3 shows the second step of the user authentication process based on a factor referring to a user's position. The second phase of the process confirms the position declared by a user.

A user starts the second phase of the geoauthentication process by applying their smartphone to a card reader (RFID interface of POS).

$$Apply\ mobile\ phone\ to\ card\ reader \tag{1}$$

The mobile application detects that the POS is nearby and receives a hello message. The MA sends the response indicating the type of emulated applet to the POS.

$$Hello\ request \tag{2}$$

$$Hello\ response \tag{3}$$

The POS obtains its own identification number POS_ID, concatenates with random number and signs it with private key KPR_POS. The random number RNDM is added to avoid the risk of a replay attack. The result of the sign operation is value SPOS_ID. The POS prepares an MPOS message consisting of signed value SPOS_ID and value POS_ID.

$$SPOS\_ID = Sign_{KPR\_POS}(POS\_ID||RNDM) \tag{4}$$

$$MPOS = (SPOS\_ID||POS\_ID) \tag{5}$$

POS sends the command that writes MPOS_ID to the mobile application. As a result MPOS is saved in memory of the mobile application.

$$Send\ MPOS \tag{6}$$

The mobile application returns a success response.

$$Success\ response \tag{7}$$

The POS sends a mobile application command that starts the geoauthentication process.

$$Geoauthenticate \tag{8}$$

The mobile application prepares messages consisting of id_1, $SUUID_G$ and MPOS, which are encrypted with the authentication service public key KP_AS. Value E_1 is signed with the mobile application private key KPR_MAS. The calculated value is stored in the mobile application memory. The result of the process is sent to the POS as a success message.

$$E1 = Enc_{KP\_AS}(id_1||SUUID_G||MPOS) \tag{9}$$

$$S1 = Sign_{KPR\_MAS}(E1) \tag{10}$$

$$M1 = (S1||E1) \tag{11}$$

$$Save\ M1\ to\ memory \tag{12}$$

$$Success\ response \tag{13}$$

The POS receives the success message and sends it to the mobile application command that reads data from the memory where signed value M1 is stored.

$$READ\ M1 \tag{14}$$

The mobile application returns a success response and transmits the bytes of value M1 to the POS.

$$return(M1) \tag{15}$$

The POS signs the received message with its own private key KPR_POS. The prepared value M2 is sent to the authentication service. The AS receives the message and returns the server response.

$$S2 = Sign_{KPR\_POS}(M1) \tag{16}$$

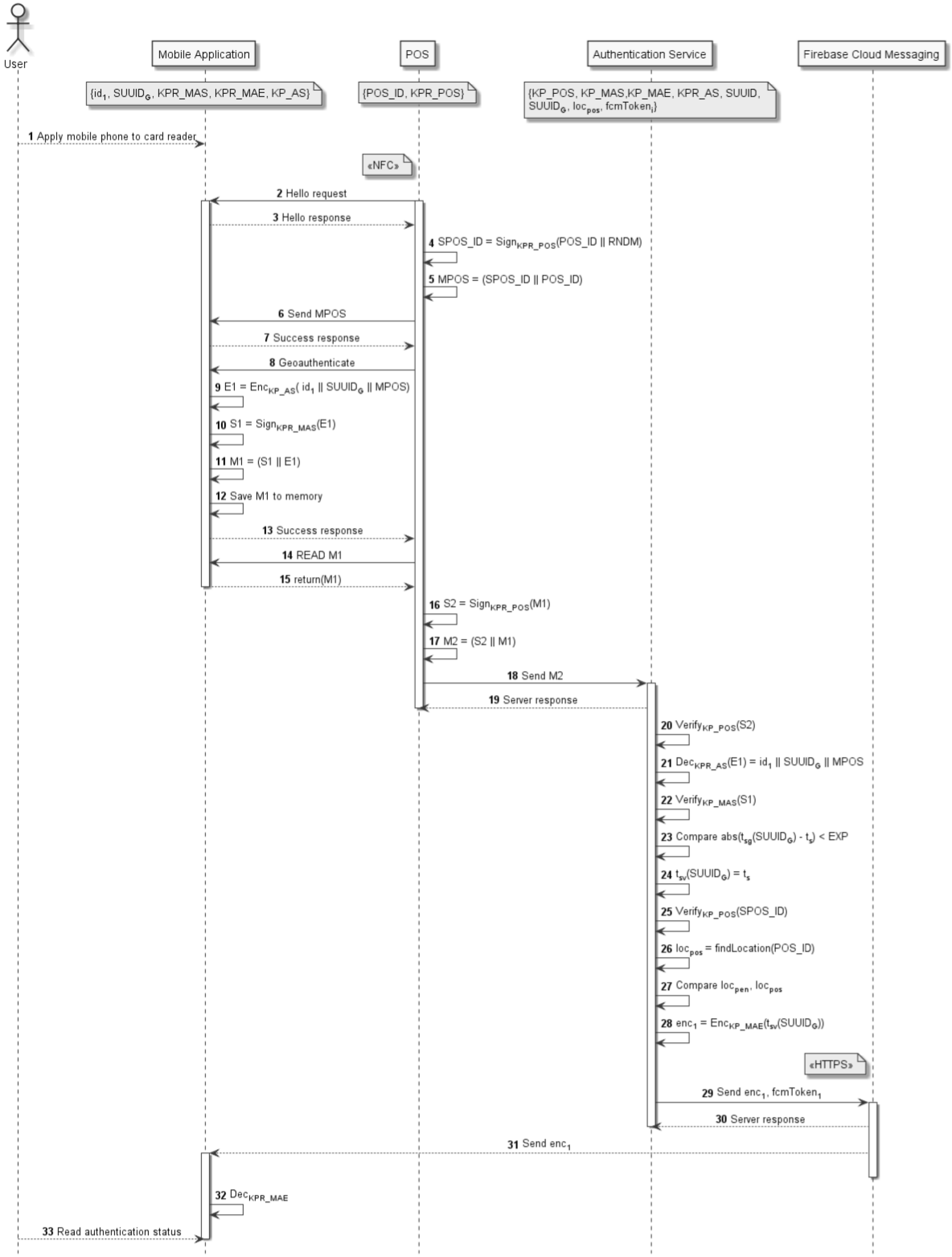

User
Mobile Application
POS
Authentication Service
Firebase Cloud Messaging
{id$_1$, SUUID$_G$, KPR_MAS, KPR_MAE, KP_AS}
{POS_ID, KPR_POS}
{KP_POS, KP_MAS,KP_MAE, KPR_AS, SUUID, SUUID$_G$, loc$_{pos}$, fcmToken$_i$}
1 Apply mobile phone to card reader
«NFC»
2 Hello request
3 Hello response
4 SPOS_ID = Sign$_{KPR\_POS}$(POS_ID || RNDM)
5 MPOS = (SPOS_ID || POS_ID)
6 Send MPOS
7 Success response
8 Geoauthenticate
9 E1 = Enc$_{KP\_AS}$(id$_1$ || SUUID$_G$ || MPOS)
10 S1 = Sign$_{KPR\_MAS}$(E1)
11 M1 = (S1 || E1)
12 Save M1 to memory
13 Success response
14 READ M1
15 return(M1)
16 S2 = Sign$_{KPR\_POS}$(M1)
17 M2 = (S2 || M1)
18 Send M2
19 Server response
20 Verify$_{KP\_POS}$(S2)
21 Dec$_{KPR\_AS}$(E1) = id$_1$ || SUUID$_G$ || MPOS
22 Verify$_{KP\_MAS}$(S1)
23 Compare abs(t$_{sg}$(SUUID$_G$) - t$_s$) < EXP
24 t$_{sv}$(SUUID$_G$) = t$_s$
25 Verify$_{KP\_POS}$(SPOS_ID)
26 loc$_{pos}$ = findLocation(POS_ID)
27 Compare loc$_{pen}$, loc$_{pos}$
28 enc$_1$ = Enc$_{KP\_MAE}$(t$_{sv}$(SUUID$_G$))
«HTTPS»
29 Send enc$_1$, fcmToken$_1$
30 Server response
31 Send enc$_1$
32 Dec$_{KPR\_MAE}$
33 Read authentication status

FIGURE 3. Confirmation of the declared user location.

$$M2 = (S2||M1) \tag{17}$$

$$Send\ S2 \tag{18}$$

$$Server\ response \tag{19}$$

The authentication service starts the verification process. Each of the verification steps performed is conducted only if the previous step has been completed. The first step checks the integrity of the received data. The AS verifies signature S2 using the POS public key KP_POS and signature S1 with the mobile application public key KP_MAS. Received encrypted message E1 is being decrypted with AS private key KPR_AS. Then, the authentication service can read the values of id_1, $SUUID_G$ and MPOS.

$$Verify_{KP\_POS}(S2) \tag{20}$$

$$Dec_{KPR\_AS}(E1) = id_1||SUUID_G||MPOS \tag{21}$$

$$Verify_{KP\_MAS}(S1) \tag{22}$$

The AS checks whether the difference between the generation time of $SUUID_G$ and the current timestamp is not greater than the assumed constant EXP. After successful validation, value t_{sv} is set to the current timestamp.

$$Compare\ abs(t_{sg}(SUUID_G) - t_s) < EXP \tag{23}$$

$$t_{sv}(SUUID_G) = t_s \tag{24}$$

The authentication service verifies the integrity of SPOS_ID using the POS public key KP_POS. Then, AS tries to find the POS location in the trusted AS registry. The trusted POS location is compared to the pending location saved during the first phase of the protocol. The AS verifies whether identity id_1 is equal to the identity to which the loc_{pen} was locked. If values are equal, the AS sets loc_{pos} as the correct user $location_r$.

$$Verify_{KP\_POS(SPOS\_ID)} \tag{25}$$

$$loc_{pos} = findLocation(POS\_ID) \tag{26}$$

$$Compare\ loc_{pen}, loc_{pos} \tag{27}$$

The authentication service tries to inform the mobile application that the authentication process succeeded. The AS prepares encrypted message enc_1 consisting of validation timestamp $t_{sv}(SUUID_G)$. The encrypted value and $fcmToken_1$ bound to user u_1's mobile application are passed to the FCM service. Firebase cloud messaging returns the server response.

$$Enc_1 = Enc_{KP\_MEA}(t_{sv}(SUUID_G)) \tag{28}$$

$$Send\ enc_1, fcmToken_1 \tag{29}$$

$$Server\ response \tag{30}$$

FCM passes encrypted message enc_1 to the mobile application. It decrypts enc_1 with the MA private key KPR_MAE; and after calculating the duration time of the process, it shows the user result of the authentication process.

$$Send\ enc_1 \tag{31}$$

$$t_{sv}(SUUID_G) = Dec_{KPR\_MAE}(enc_1) \tag{32}$$

The user can read the authentication status.

$$Read\ authentication\ status \tag{33}$$

IV. PROTOCOL EVALUATION

A. SECURITY CONSIDERATIONS

Research experiments based on the presented protocol were conducted. The experiments implemented several possible attack scenarios related to the use of location data as an authentication factor. The results of the experiments and analyses are presented below and collected in Table 3.

1) GPS LOCATION SPOOFING ATTACK

The use of location data to provide security services creates an additional attack risk. Information about a user's current location may be considered sensitive and may be intercepted. Moreover, location information obtained from a smartphone can be manipulated. The presented authentication protocol based on location data has been designed to minimize the risks resulting from the use of location data. In order to eliminate the possibility of conducting an attack consisting of modifying information about a user's location, methods securing the process on two levels were applied. The first level is related to the mobile application, and the second is the protocol level. First, the mobile application obtains location information on the basis of three independent interfaces: GPS signals, data from the GSM provider and an external service that uses information about the IP addresses of Wi-Fi networks available nearby. In order to conduct the authentication process, it was assumed that at least two sources of location data (e.g., GPS and GSM) should be used. In addition, the protocol reduces the risk of an attack using location data modifications by acquiring and comparing location data from various devices - user u_i's smartphone, devices of other users confirming user u_i's identity and a POS terminal located in a specific place where the authentication process is performed. Therefore, an effective attack on authentication based on the location-based factor would have to be conducted on the three elements of the presented protocols of the MA, WA and POS. One of the attack scenarios assumes modifications to the location data of user ui. Suppose in this case that the attacker is able to force the GPS module to return specific coordinates. Then, the location determined on the basis of the GSM network and Wi-Fi network is inconsistent with the GPS data. In such a situation, data manipulation will be detected. Moreover, according to the assumptions, the authentication process based on two sources of position data is possible. Therefore, such a scenario does not prevent the execution of the user identity confirmation process. The second scenario assumed increasing the attacker's potential and that the attacker is able to influence the information returned based on the GSM module and Wi-Fi network. Additionally, in this case, the user authentication process will not occur. The manipulated location data of user u_i will not match the data obtained from the user's witness application and the POS location at which the process is performed. Another attack scenario assumed that the attacker is able to modify the location data on user u_i's device and can modify the location data on the device on which the confirming application is installed (the witness application). In this case, the security of the authentication process can be ensured by increasing the number of required confirmations. Assuming that the location data on user u_i's device and on several instances of the witness application are manipulated, the authentication process will not occur. The fraud will be detected because the information obtained from the user's application will not match the location obtained from the other, unattacked witness application instances. This solution will increase the process execution time but will provide a higher level of security for the factor based on location data. If we increase the attacker's potential, we can assume that all packets from the acknowledgement devices ($M_1, M_2, \ldots, M_N$) sent to the POS terminal and to the authentication server have been modified and the coordinates sent in them are consistent with each other. In this case, the authentication process will also fail because the attacker is unable to modify the information about the POS location. The attack on the mobile application and the witness application allows for the manipulation of only the data sent in the first phase of the declaration of the user's location. The second phase, confirming the user's location, is conducted based on a trusted POS device whose location is known and verified based on cryptographic methods.

2) LOCATION PRIVACY

In the case of location privacy, the scenario that has been tested assumed the possibility of tracking a user and a case of data leakage or interception. It seems that the proposed scheme also considers the requirements of ensuring the confidentiality of the location data used in the authentication process. At each step of the protocol implementation, the location data sent to the next node are encrypted with the public key of the authentication service. As a result, only the AS is able to read the location of a particular device. This solution makes it impossible to track devices; and in the case of data leakage or interception, the devices cannot be read.

3) MAN-IN-THE-MIDDLE ATTACK

Another verified scenario is a man-in-the-middle attack. In this case, an attack was unsuccessful because the cryptographic key pairs were exchanged between the authentication service and the other units involved in the communication. Each of the messages between the POS, mobile application, witness application and authentication service is signed. Therefore, an attempt to implement an MITM attack at any stage of the protocol implementation will be detected on the side of the authentication service. The security of the protocol in terms of an MITM attack has also been formally verified, and the results of this analysis are presented in Chapter 4.B.

4) REPLY ATTACK

The last of the examined scenarios relied on conducting a reply attack. In this case, the attack failed because pseudorandom numbers were added to the protocol to protect against a reply attack. Each of the exchanged messages contains an RNDM component that is added to the message before it is encrypted. This allows you to verify if the messages sent to the server were not intercepted by the intruder and used again for a reply attack. After decrypting a message, the random number is verified. If this number is repeated, the query is not processed. A formal protocol security analysis in terms of a replay attack was conducted, and the results are presented in Chapter 4.B.

TABLE 3. Comparison with other existing schemes.

Criteria	[29]	Ours
GPS location spoofing attack	Yes	Yes
Location privacy	No	Yes
Man-in-the-middle	Yes	Yes
Reply attack	No	Yes

B. PERIOD FORMAL VERIFICATION OF THE CORRECTNESS AND SAFETY OF THE PROTOCOL

Since the 1990s, researchers have proposed several methodologies for protocol property verification. In addition to simple real or virtual simulation, several mathematical methods, including inductive and deductive methods, were proposed [33], [34]. Such methods allow formal proof of whether the considered protocol possesses an investigated security property [33]. Today, the most efficient methods in this area are model checking of transition systems that encode users' behaviours, including their knowledge, during a protocol's executions [35]–[41]. For several years, it has been crucial to consider the properties of time-dependent protocols. It is not easy to consider rather complicated time models. There are only a few well-grounded approaches that allow investigation of the time dependencies between protocols' parameters or users' behaviours [42]–[44].

Additionally, as the protocols are expanded with new security techniques, such as location in our case, there is a need to find current verification methods. The consistency, internal correctness, and the probability of an attack occurring upon the authentication and confidentiality properties of the protocol will be formally examined. Here, we propose formal multilevel verification of the proposed protocol. First, we examine the untimed version of the protocol seeking attacks of malicious intruders upon authentication and secrecy properties. After concluding that the untimed version is safe, we proceeded to consider a time protocol version. Here, we can strictly compute the minimal lifetime values that allow protocol execution and compute maximal values that guarantee the protection of the protocol against all undesirable, malicious man-in-the-middle behaviours. In our investigation, we consider the well-known and most commonly used Dolev-Yao model to verify an intruder [45].

Starting our work, we must initially make some assumptions. First, we assume the perfect cryptography assumption where an intruder will not be able to break the cipher without knowing the key. We also assume that users and the server have a set of locations considered to be appropriate for a specific user, which are, in some sense, its identifiers. The server has the necessary keys, passwords, and data used for communication. We also assume that users have a mobile device equipped with private keys and a biometric reader.

For formal protocol verification using a particular verification tool, we need to describe the specification languages used by the tool. Here, we must apply strict language rules that make it impossible to apply descriptions used before to determine, for example, protocol messages. First, we present the protocol specified in the so-called *Alice-Bob notation*. In this case, for the subsequent considerations, a symbol A denotes the user who is being authenticated (that uses the mobile application), S denotes the authentication service (server), B denotes the proving application, and P denotes the point-of-sale terminal.

1. $A \rightarrow S$: $\langle i(A)|N_A\rangle_{K_{AS}}$
2. $S \rightarrow A$: $\langle t_S\rangle_{K_{AS}}$
3. $A \rightarrow S$: $\langle t_S\rangle_{K_{AS}}$
4. $S \rightarrow A$: $\langle N_S|t_S\rangle_{K_A^+}$
5. $A \rightarrow S$: $\langle N_S\rangle_{K_{AS}}$
6. $S \rightarrow A$: $\langle t'_S\rangle_{K_{AS}}$
7. $A \rightarrow S$: $\langle t'_S\rangle_{K_{AS}}$
8. $S \rightarrow A$: $\langle\langle N'_S\rangle_{K''^+_A}|t_{N'_S}\rangle_{K'^+_A}$
9. $A \rightarrow S$: $\langle N'_S|t_{N'_S}\rangle_{K_S^+}$
10. $A \rightarrow B$: $\langle\langle N'_A|t_{N'_A}\rangle_{K_S^+}\rangle_{K_A^-}$
11. $B \rightarrow A$: $\langle\langle\langle\langle N'_A|t_{N'_A}\rangle_{K_S^+}\rangle_{K_A^-}|N_B|t_{N_B}\rangle_{K_S^+}\rangle_{K_B^-}$
12. $A \rightarrow S$: $\langle\langle\langle\langle N'_A|t_{N'_A}\rangle_{K_S^+}\rangle_{K_A^-}|N_B|t_{N_B}\rangle_{K_S^+}\rangle_{K_B^-}$
13. $A \rightarrow P$: $i\,(A)$
14. $P \rightarrow A$: $\langle N_P\rangle_{K_P^-}$
15. $A \rightarrow P$: $\langle\langle i(A)|t_A|\langle N_P\rangle_{K_P^-}\rangle_{K_S^+}\rangle_{K_A^-}$
16. $P \rightarrow S$: $\langle\langle\langle i(A)|t_A|\langle N_P\rangle_{K_P^-}\rangle_{K_S^+}\rangle_{K_A^-}\rangle_{K_P^-}$
17. $S \rightarrow A$: $\langle N_P\rangle_{K_P^+}$

In this notation, N and t denote nonces and timestamps, respectively; $\langle X\rangle_K$ denotes a message X encrypted with the key K; and $X|Y$ is the concatenation of X and Y. K_A^+ and K_A^- denote the public and private keys of A, respectively. In the considered protocol, there are several messages with multilevel encryption. An example is $\langle\langle\langle i(A)|t_A|\langle N_P\rangle_{K_P^-}\rangle_{K_S^+}\rangle_{K_A^-}\rangle_{K_P^-}$, where there is message $i(A)|t_A|\langle N_P\rangle$ first signed by private key K_P^-, then encrypted by public key K_S^+ and next signed two times with private keys with K_A^- and K_P^-.

As we can see, in this way, we can strictly specify each of the protocol's steps. Using such notations, we can start formal analysis and specify all the protocol's steps in other, more complicated languages connected with suitable verification tools.

The practice of protocol verification techniques shows that at the beginning of the study of the correctness of the protocol, usually an untimed version of the protocol should be examined. For this purpose, we chose AVISPA [35] due to the popularity and capabilities of this tool. AVISPA was the result of a large project cocreated by scientists and industry. It consists of four modules, each of which can test the protocol differently. It requires the specification of a protocol in HLPSL language [35]. Below we illustrate a specification of role A:

```
role alice (A, B, P, S: agent,
Ka, K1a, K2a, Kb, Ks, Kp: public_key,
inv(Ka), inv(Kb), inv(Kp): private_key,
Kas: symmetric_key,
SND, RCV: channel (dy))
played_by A def=
  local State: nat,
Na, N1a, N2a, N3a, Ns, N1s, N2s,N3s,
N4s, N5s, Nb, N1b, Np: text
  init State: = 0
  transition
0. State = 0 /\ RCV(start) =|>
   State': = 1 /\ Na': = new() /\
SND({Na'.A}_Kas)
           /\ secret(Na',na,{A,S})
           /\
witness(A,S,server_alice_na,Na')
4.  State = 4 /\ RCV({N2s'}_Kas) =|>
     State': = 5 /\ SND({N2s'}_Kas)
8.  State = 8 /\ RCV({Ns'.N1s'}_Ka) =|>
     State': = 9 /\ SND({Ns'}_Kas)
12.  State = 12 /\ RCV({N3s'}_Kas) =|>
     State': = 13 /\ SND({N3s'}_Kas)
16.  State = 16 /\
RCV({{N4s'}_K2a.N5s'}_K1a) =|>
State': = 17 /\ N1a': = new() /\ N2a': =
new() /\ SND({N4s'.N5s'}_Ks) /\
SND({{N1a'.N2a'}_Ks}_inv(Ka))
          /\ secret(N1a',n1a,{A,S})
          /\ secret(N2a',n2a,{A,S})
          /\
witness(A,S,server_alice_n1a,N1a')
          /\
witness(A,S,server_alice_n2a,N2a')
24.   State = 24 /\ RCV({Np'}_inv(Kp)) =|>
     State': = 25 /\ N3a': = new() /\
SND({{A.N3a'.{Np'}_inv(Kp)}_Ks}_inv(Ka))
          /\ secret(N3a',n3a,{A,S})
          /\
witness(A,S,server_alice_n3a,N3a')
25. State = 30 /\ RCV({Np}_Ka) =|>
    State': = 31 /\
request(A,S,server_alice_n2s,N2s)
    /\ request(A,S,server_alice_ns,Ns)
    /\ request(A,S,server_alice_n1s,N1s)
    /\ request(A,S,server_alice_n3s,N3s)
    /\ request(A,S,server_alice_n4s,N4s)
end role
```

Because we are investigating the authentication properties between A and S and secrecy properties of the nonces used, we have the following security goals written in HLPSL:

```
secrecy_of na, n1a, n2a, ns, n1s, n2s, n3s,
n4s, n5s, nb, n1b, np
authentication_on alice_server_ns
authentication_on alice_server_n1s
authentication_on alice_server_n2s
authentication_on alice_server_n3s
authentication_on alice_server_n4s
```

After several minutes of computations, the AVISPA tool reported that the untimed version of the protocol is consistent, internally correct and safe according to attacks upon authentication and the security of confidential data.

In the case of time-dependent properties, there are at least two types of problems. The first is the protocol's vulnerability to replay attacks. The second is the protocol's vulnerability to malicious man-in-the-middle behaviour when a passive intruder can only retransmit data sent between honest users [44]. Such behaviour cannot be treated as a real attack upon the protocol because the authentication process is not corrupted and secret data are not compromised, but such behaviour is at least undesirable in the network. The lifetime values for subsequent timestamps should be chosen well to protect the protocol against such vulnerabilities. To solve these problems, we need to use another formalism and methodology than presented previously.

Among the methods currently leading is timed model checking, which is efficient and possesses the ability to use many different formal models. In our study, we use some type of transition systems model based on a network of synchronized timed automata in which the tested values will be checked through the reachability of specific states [38].

For our tool, we need the specification in ProToc language that allows for the description of time-connected dependencies [40]. Such a specification method allows this description and investigates two types of time primitives: time values of many timestamp generations and lifetime values. Using this, we can investigate the possibility of an attack as a function of time dependencies between the aforementioned values.

As an example of the protocol's steps specification, we present the first four steps of the protocol:

- `p_1,p#1;i(p_1),n_1(1),k_1#1(1);n_1(1);true; <k_1#1(1),i(p_1)|n_1(1)>;`
- `p#1,p_1;s#1(1),k_1#1(1);s#1(1);c[s#1(1)]; <k _1#1(1),s#1(1)>;`
- `p_1,p#1; <k_1#1(1),s#1(1)>;c[s#1(1)]; <k_1#1 (),s#1(1)>;`
- `p#1,p_1;n#1(1),s#1(2),k_1#1(1);n#1(1),s#1(2); c[s#1(1)]; <k_1#1(1),n#1(1),s#1(2)>;`

Here, we can see that each step consists of five sections. In the first of them, we can find the sender and the recipient of the step. The second section consists of all primitives that are necessary to create the message. The third section includes the primitives that must be generated for the current step. The fourth section consists of a time condition. In the last section, we can find a message that will be sent in the current step.

Let us analyse the first and second steps of our protocol described in ProToc [46].

The sender in the first step is A (p_1), and the recipient is the trusted server (p#1). The sender needs three primitives to compose the message: A's identifier (i(p_1)), A's nonce (n_1(1)) and a symmetric key shared between A and server (k_1#1(1)). At the beginning of this step, the sender does not have the nonce (n_1(1)), so he must generate this primitive. Such information is contained in the third section. There are no time conditions imposed on this step, so the fourth section informs us that all time conditions are met (true). In the last section, we can find a message (<k_1#1(1),i(p_1)|n_1(1)>) created with primitives from the second section of the specification.

The sender in the second step is the trusted server (p#1), and the recipient is A (p_1). The sender needs two primitives to compose the message: the server's timestamp (s#1(1)) and a symmetric key shared between A and the server (k_1#1(1)). The server must generate its timestamp (third section). In this step, the time condition assigned as c[s#1(1)] is imposed. Such notation denotes the time condition $\tau_S < L_f$, where τ_S is the period from the ticket generation time t_S to the current moment in the considered protocol execution, and L_f is a lifetime value of timestamp t_S. In short, this means that assumptions of the time condition must be compared with the timestamp (s#1(1)). In the last section, we can find a message (<k_1#1(1),n#1(1),s#1(2)>) created with primitives from the second section of the specification. The remaining steps should be considered in the same way.

Now, let us briefly introduce the formal model used for verification. Protocol modelling and verification using networks of synchronized automata were introduced in [38]. In such works, there are two types of automata in the network: those representing protocol executions and those representing users' knowledge. Knowledge research causes unrealistic steps to be excluded from the entire network, which is impossible to perform due to the lack of knowledge of the user, server, or intruder about the elements required in the step. Execution automata allow synchronization and ensure the correctness of the steps interlaced in different executions of the same protocol.

The synchronized automata network finally creates the so-called product automaton, which is encoded into a Boolean propositional formula. The formula, due to its size, is tested using the SAT solver. Then, we can check the reachability of the states equivalent to an attack. The tool then verifies the timestamps. Finally, we simulated the latency times in the network using different probability distributions: uniform, normal, Poisson, Cauchy and exponential. For the tests, we used a computer with the Linux Ubuntu operating system, an Intel Core i7 processor and 16 GB RAM.

In the case of tests with the SAT solver, we received the result UNSAT, which means that the protocol is safe. The SAT solver worked with path lengths equal to 70,62516 clauses and 146540 literals and used 10.29 MB of memory. The study lasted 21.9515 s.

Next, we performed timed analysis of executions (TAoE). First, we assumed the following:

- The encryption and decryption times were equal to 2 tu.
- The generation time of confidential information was equal to 1 tu.
- The range of delay in the network was equal <1 tu, 5 tu>.

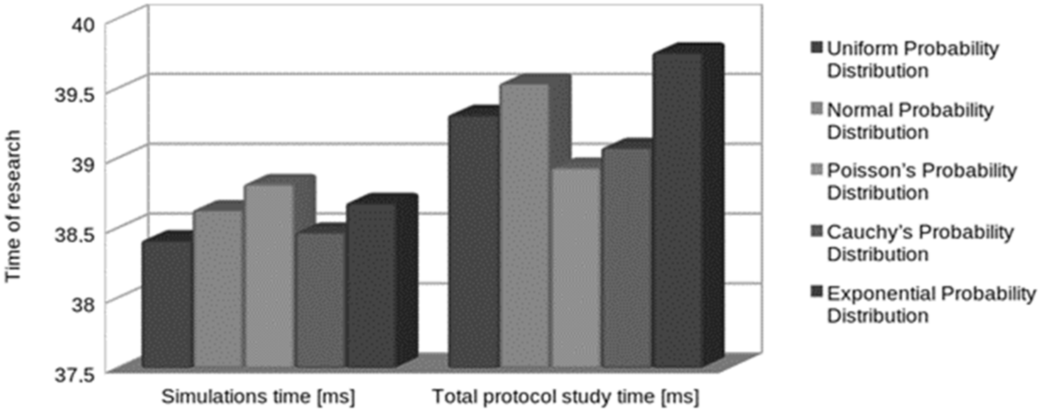


FIGURE 4. Time of research for SoE.

TABLE 4. Lifetimes values in the following steps for TAoE and SoE in [tu].

Step	Lifetime for TAoE	Lifetime for SoE	Step	Lifetime for TAoE	Lifetime for SoE
1	179	264	10	87	127
2	169	249	11	72	107
3	159	234	12	57	87
4	150	220	13	20	77
5	139	204	14	47	67
6	130	190	15	37	52
7	120	175	16	23	33
8	111	161	17	9	14
9	96	141			

- The current delay in the network value was equal to 1 tu.

Next, we calculated minimal session time (111 tu), maximal session time (179 tu) and lifetime in the following steps. The lifetime values for TAoE are presented in Table 4. During the calculation of the lifetime values, the values of encryption and decryption times, generation of confidential information and network delays were considered.

We obtained the following results:

- Structure creation time: 0.043302 ms
- Time of timed analysis: 0.380163 ms
- Total protocol study time: 0.427735 ms

During this research phase, the duration of all executions was analysed. Only fair execution proved to be feasible. In the others, the intruder could not acquire the appropriate knowledge by seeking to take additional steps. These executions failed in 'failure to meet the imposed time' conditions.

Next, we performed simulations of executions (SoE). First, we assumed the following:

- The encryption and decryption times were equal to 2 tu.
- The generation time of confidential information was equal to 1 tu.
- The range of delay in the network was equal <1 tu, 5 tu>.

The current value of the delay in the network was randomly generated according to the mentioned probability distributions.

Next, we calculated the minimal session time (111 tu), maximal session time (264 tu) and lifetime in the following steps. The lifetimes values for TAoE are presented in Table 4.

We show the obtained timed results in Figure 4. During this phase of the research, simulations of all executions were conducted considering the random delay in the network values. These values were randomized according to selected probability distributions (uniform, normal, Poisson, Cauchy and exponential). Each execution has been tested in 100 test series. Similar to the analysis of the times, only fair executions proved possible. In the others, the intruder could not acquire the appropriate knowledge by seeking to take additional steps. These executions failed to meet the imposed time conditions.

The results presented in this section proved that the proposed authentication protocol is consistent, internally correct and safe from authentication and secrecy attacks. Additionally, if the proposed values of timestamp lifetimes are used, the protocol is safe from replay attacks and the man-in-the-middle malicious behaviour.

C. PERFORMANCE

Designing an authentication protocol is a complex and complicated issue because it is related to user interaction. In addition to confirming the effectiveness and safety of the presented protocol, its performance and the feasibility of implementation in a dedicated environment should be verified. In order to test the performance of the proposed protocol, we prepared a test model that performs multifactor authentication based on the presented scheme [14] and the location-based authentication factor proposed in the article. The aim of the conducted experiments was to evaluate whether the time of the authentication process, based on

TABLE 5. Results of the performance evaluation.

Statistical metrics	Values	Percentage
The entire authentication process, average time	14599.87 ms	100%
Possession factor, average time	7494.70 ms	51%
Inherence factor, average time	2560.85 ms	18%
Location-based factor, average time	4544.31 ms	31%

the factor using location data in the manner presented in the protocol, differs from the authentication steps using the factors of other categories. The determination of the efficiency of the factor was made by comparing the duration of the authentication with the duration of the authentication based on the possession and inherence factors. The prepared model implements the architecture shown in Figure 1. The possession factor was implemented as an OTP code sent as a push notification, and the inherence factor was based on the fingerprint in accordance with the information presented in previous studies [14]. The model was extended with the witness application and a module in the mobile application, which implemented communication within the presented protocol (Figure 2 and Figure 3). Additionally, a POS emulator was prepared for the second phase (Figure 3). The results of the conducted research are presented in Table 5. The tests were conducted in an environment running on a public internet network. The average values were determined on the basis of n = 200 samples, which determines the number of attempts made in the multifactor authentication process. Each of the authentication processes was performed by the user without automating any of the steps. Such tests were designed to reproduce the authentication process in real conditions. The architecture of the prepared model is the environment in which the presented protocol could be implemented.

The average authentication time based on location data ignores the waiting time for determining a location based on GPS or other location sources because the goal is to compare the implementation of individual authentication steps, not the speed of individual interfaces of the device. The time for obtaining the location was also omitted because it depends on the environment in which the user is located. The presented results indicate that the use of the factor based on location data may be more efficient and faster than the implementation of the authentication factor based on the OTP code sent to the user. The obtained results confirm that the use of the proposed authentication factor does not significantly affect the duration of the authentication process. The next stage of the research is to compare the performance of the multifactor authentication protocol [14] extended with the proposed scheme using the factor based on location data. For this purpose, the existing prototype has been parameterized in such a way that the authentication process can be performed based on selected authentication factors. The conducted studies included the following sets of authentication factors: {K, I}, {K, L}, {I, L}, {K, P}, {P, I}, {P, L}, {K, L, I}, {K, P, I}, {K, P, L}, {L, P, I}, and {K, P, I, L}, where "K" is a knowledge-based factor, "P" is a factor from the possession category, "I" is a factor from the inherence category, and "L" is a factor implemented on the basis of location data, as shown in Figure 2 and Figure 3. The following diagram, Figure 5, presents a comparison of the authentication process performance for different sets of authentication factors. The presented results show that in a situation where the location-based factor has replaced a factor from another category, there is no significant change in the duration of the authentication process. The differences between the various sets of authentication factors used are relatively small.

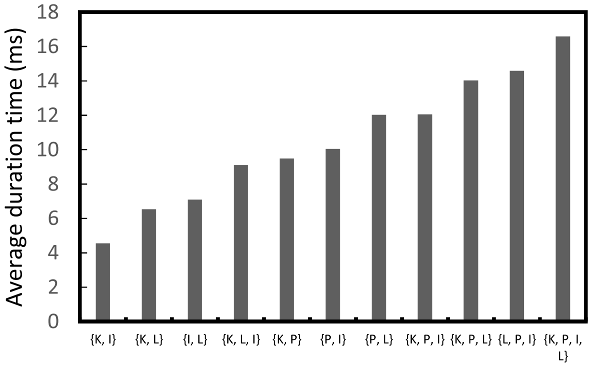


FIGURE 5. Performance comparison of the authentication factor sets. K – knowledge, P – possession, I – inherence.

V. DISCUSSION

The presented protocol uses a user's location data in the authentication process and can be used in the multifactor authentication process or as an additional factor supporting the existing multifactor authentication systems. In order to determine whether the location-based factor can be treated as an authentication factor, it is necessary to verify the compliance of the proposed solution with the specific requirements related to the authentication factor (Table 1). The first requirement is related to the verification of the user's identity. The proposed protocol signs the location data using user u_i's private key. During the verification of the received data, the authentication service is able not only to verify the integrity of the data received but also to confirm the identity of the user who sent the data. The presented authentication scheme meets the requirement according to which the authentication factor must be controlled by the user. This feature is ensured by the private key property that was generated in the user's mobile device system keystore. The private key property [47] allows the key to be used only after the user has been authenticated. Therefore, only the real smartphone user who controls all authentication factors is able to release the authentication factor based on location data. The third issue is related to the association of the authentication factor with the user identity, and this is an issue that has not been clearly described in previous publications on location-based authentication. It is hard to assume that the location will be associated with a specific user, similar to other authentication factors that are related to the identity of a specific user. The limitation of the factor based on location data allows us to solve this problem in two ways. The first approach assumes that a specific location can only be assigned to one user, provided that only one user has access to a given area. In a one-to-one relation, an authentication system receiving data from a particular location is able to determine the identity of the user being authenticated. Considering the operating conditions of a real system, the implementation of such an assumption is almost impossible. The second approach assumes adjusting this requirement to the conditions in such a way that the implementation of authentication based on the location-based factor is possible in any place. In this case, it should be assumed that a certain area is assigned to a specific user identity. The implementation of this requirement consists of assigning an area to a user and, at the same time, blocking the possibility of another user using this area for a specified period of time. The data obtained in this way are compared with a trusted location obtained from another source, which is a resistant to attempts to modify the location data (e.g., from a POS terminal). As a result, the location-based factor meets the requirements of the authentication factor that must be associated with the user. The last assumption is related to the issue of comparing the reference data to the data provided during the authentication process, as in the case of authentication based on factors from the knowledge, possession and inherence groups. This process is also similar when using the factor based on location data. The difference is that the pattern data are not transferred during the registration process and are transferred during the first phase of the authentication process (Figure 2). The presented results of the analysis indicate that the proposed authentication factor using location data meets the requirements for authentication factors. The most important step in designing the location-based authentication factor is the assumption of assigning a specific location to only one identity at a time. This assignment can be done in two ways: statically and dynamically. In the case of a static assignment of an area to an identity, only one person can be authenticated in a given area. In the case of dynamic assignment, a given location is associated with the identity of the user for a specified period of time and may be associated with the identity of another user after the end of the process. If the time condition is not met and a given area is not associated only with the identity of one user, we would not be able to identify the identity of two users standing next to each other. Therefore, it can be concluded that location may be a feature assigned to a user only at a specific time. Moreover, in order to perform the authentication process based on the location data, it is necessary to possess location data from two different sources. The data from the first source come from a device associated with the user and act as challenge data that are compared with the reference data. The data from the second source are the reference data and are determined by the trusted device. The methods used to determine a user's location data should be resistant to attacks, including modification attempts. This requirement has been met by the use of appropriate mechanisms during location determination and at the protocol design level (Chapter IV. A).

VI. CONCLUSION

This article presents a new authentication protocol that uses a user's location data in the user authentication process. The requirements of the authentication process and collection of the features that characterize the authentication process and authentication factors have been presented. These features have been analysed in detail because location data are very often used in schemes that perform the authorization task and not user authentication. Several publications related to the use of location data for the implementation of security services in a mobile environment have been referenced, and other methods of user authentication in a mobile environment have been discussed. Based on this detailed analysis, a user authentication protocol based on location data has been proposed. It should be emphasized that according to the authors' best knowledge, the authentication process based on location data is presented here for the first time. The process must be conducted in two steps and should consist of the phase of declaring the user's location and confirming the declared location. The presented scheme has been verified in terms of authentication performance based on the location-based factor. Its performance was compared to the systems using other factors from the knowledge, possession and inherence categories. The conducted research has been experimentally verified. The implementation allowed the configuration of factors that were used for authentication in a given test scenario. Moreover, security analysis, which used the prepared security tests and allowed formal verification of the protocol's resistance to man-in-the-middle attacks and reply attacks, was conducted. The results indicate that the proposed scheme uses location data in a way that allows the data to be treated as an authentication factor associated with a user. The authors of the article recommend using the location-based factor as an alternative to traditional authentication factors when it is not possible to use multifactor authentication based on knowledge, possession or inherence. Alternatively, the proposed scheme based on location-based factors can be used as an additional authentication factor, increasing the security of the identity confirmation process in a mobile environment.

REFERENCES

[1] P. A. Grassi, "Digital identity guidelines," NIST Special Publication, Gaithersburg, MD, USA, Tech. Rep. NIST Special Publication 800-63-3, Jun. 2017, p. 75.

[2] (Nov. 2015). *Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on Payment Services in the Internal Market, Amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU), No 1093/2010, and Repealing Directive 2007/64/EC (Text With EEA Relevance)*, Official Journal of the European Union. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32015L2366

[3] *Commission Implementing Regulation (EU) 2015/1502 of 8 September 2015 on Setting out Minimum Technical Specifications and Procedures for Assurance Levels for Electronic Identification Means Pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market*, Official J. Eur. Union, Brussels, Belgium, Sep. 2015, p. 14.

[4] C. Stephens, ''Why are SMS codes still the global ID solution?'' *Biometric Technol. Today*, vol. 2020, no. 8, pp. 8–10, Sep. 2020, doi: 10.1016/S0969-4765(20)30110-7.

[5] S. Plaga, M. Niethammer, N. Wiedermann, and A. Borisov, ''Adding channel binding for an Out-of-Band OTP authentication protocol in an industrial use-case,'' in *Proc. 1st Int. Conf. Data Intell. Secur. (ICDIS)*, Apr. 2018, pp. 250–257, doi: 10.1109/ICDIS.2018.00048.

[6] E. Erdem and M. T. Sandikkaya, ''OTPaaS—One time password as a service,'' *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 743–756, Mar. 2019, doi: 10.1109/TIFS.2018.2866025.

[7] (Apr. 12, 2019). *Regulatory Technical Standards on Strong Customer Authentication and Secure Communication Under PSD2*. European Banking Authority. Accessed: Dec. 18, 2021. [Online]. Available: https://www.eba.europa.eu/regulation-and-policy/payment-services-and-electronic-money/regulatory-technical-standards-on-strong-customer-authentication-and-secure-communication-under-psd2

[8] N. R. Kisore and S. Sagi, ''A secure SMS protocol for implementing digital cash system,'' in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Aug. 2015, pp. 1883–1892, doi: 10.1109/ICACCI.2015.7275893.

[9] S. Holtmanns and I. Oliver, ''SMS and one-time-password interception in LTE networks,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6, doi: 10.1109/ICC.2017.7997246.

[10] N. D. Sarier, ''Multimodal biometric authentication for mobile edge computing,'' *Inf. Sci.*, vol. 573, pp. 82–99, Sep. 2021, doi: 10.1016/j.ins.2021.05.036.

[11] I. Stylios, S. Kokolakis, O. Thanou, and S. Chatzis, ''Behavioral biometrics & continuous user authentication on mobile devices: A survey,'' *Inf. Fusion*, vol. 66, pp. 76–99, Feb. 2021, doi: 10.1016/j.inffus.2020.08.021.

[12] Q. Feng, D. He, S. Zeadally, and H. Wang, ''Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment,'' *Future Gener. Comput. Syst.*, vol. 84, pp. 239–251, Jul. 2018, doi: 10.1016/j.future.2017.07.040.

[13] T. Phillips, X. Yu, B. Haakenson, S. Goyal, X. Zou, S. Purkayastha, and H. Wu, ''AuthN-AuthZ: Integrated, user-friendly and privacy-preserving authentication and authorization,'' in *Proc. 2nd IEEE Int. Conf. Trust, Privacy Secur. Intell. Syst. Appl. (TPS-ISA)*, Oct. 2020, pp. 189–198, doi: 10.1109/TPS-ISA50397.2020.00034.

[14] B. Maciej, E. F. Imed, and M. Kurkowski, ''Multifactor authentication protocol in a mobile environment,'' *IEEE Access*, vol. 7, pp. 157185–157199, 2019, doi: 10.1109/ACCESS.2019.2948922.

[15] *Minimum Security Requirements for Federal Information and Information Systems*, Nat. Inst. Standards Technol., NIST FIPS, Gaithersburg, MD, USA, Mar. 2006, doi: 10.6028/NIST.FIPS.200.

[16] P. Bethi, S. Pathipati, and A. P, ''Stealthy GPS spoofing: Spoofer systems, spoofing techniques and strategies,'' in *Proc. IEEE 17th India Council Int. Conf. (INDICON)*, Dec. 2020, pp. 1–7, doi: 10.1109/INDICON49873.2020.9342317.

[17] B. MacRae, A. Salehi-Abari, and J. Thorpe, ''An exploration of geographic authentication schemes,'' *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 1997–2012, Sep. 2016, doi: 10.1109/TIFS.2016.2570681.

[18] H. Alamleh and A. A. S. AlQahtani, ''Architecture for continuous authentication in location-based services,'' in *Proc. Int. Conf. Innov. Intell. Informat., Comput. Technol. (ICT)*, Dec. 2020, pp. 1–4, doi: 10.1109/3ICT51146.2020.9311972.

[19] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, ''Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location,'' *IEEE Syst. J.*, vol. 11, no. 2, pp. 513–521, Jun. 2017, doi: 10.1109/JSYST.2015.2472579.

[20] D. M. Shila and K. Srivastava, ''CASTRA: Seamless and unobtrusive authentication of users to diverse mobile services,'' *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4042–4057, Oct. 2018, doi: 10.1109/JIOT.2018.2851501.

[21] Z. Dou, I. Khalil, and A. Khreishah, ''A novel and robust authentication factor based on network communications latency,'' *IEEE Syst. J.*, vol. 12, no. 4, pp. 3279–3290, Dec. 2018, doi: 10.1109/JSYST.2017.2691550.

[22] B. Akoramurthy and J. Arthi, ''GeoMoB—A geo location based browser for secured mobile banking,'' in *Proc. 8th Int. Conf. Adv. Comput. (ICoAC)*, Jan. 2017, pp. 83–88, doi: 10.1109/ICoAC.2017.7951750.

[23] M. Ahmadi and B. S. Ghahfarokhi, ''Preserving privacy in location based mobile coupon systems using anonymous authentication scheme,'' in *Proc. 13th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2016, pp. 60–65, doi: 10.1109/ISCISC.2016.7736452.

[24] W. Wang, Y. Chen, and Q. Zhang, ''Privacy-preserving location authentication in Wi-Fi networks using fine-grained physical layer signatures,'' *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1218–1225, Feb. 2016, doi: 10.1109/TWC.2015.2487453.

[25] G. Sun, S. Cai, H. Yu, S. Maharjan, V. Chang, X. Du, and M. Guizani, ''Location privacy preservation for mobile users in location-based services,'' *IEEE Access*, vol. 7, pp. 87425–87438, 2019, doi: 10.1109/ACCESS.2019.2925571.

[26] O. AlHory, O. Shoushara, H. Al Suri, M. Al Shunnaq, and F. Awad, ''5G mmWave indoor location identification using beamforming and RSSI,'' in *Proc. 11th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2020, pp. 091–095, doi: 10.1109/ICICS49469.2020.239532.

[27] *Information Technology—Radio Frequency Identification for Item Management—Part 3: Parameters for Air Interface Communications at 13, 56 MHz*, Standard ISO/IEC 18000-3:2010(en), Accessed: Jul. 14, 2021. [Online]. Available: https://www.iso.org/obp/ui/#iso: std: iso-iec:18000:-3: ed-3: v1: en

[28] A. Adukkathayar, G. S. Krishnan, and R. Chinchole, ''Secure multifactor authentication payment system using NFC,'' in *Proc. 10th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Jul. 2015, pp. 349–354, doi: 10.1109/ICCSE.2015.7250269.

[29] C.-M. Chen, X. Zhang, and T.-Y. Wu, ''A secure condition-based location authentication protocol for mobile devices,'' in *Proc. 3rd Int. Conf. Comput. Meas. Control Sensor Netw. (CMCSN)*, May 2016, pp. 146–149, doi: 10.1109/CMCSN.2016.16.

[30] *Firebase Cloud Messaging*. Firebase. Accessed: Jul. 28, 2021. [Online]. Available: https://firebase.google.com/docs/cloud-messaging

[31] P. J. Leach, M. Mealling, and R. Salz. *A Universally Unique IDentifier (UUID) URN Namespace*. Accessed: Feb. 2, 2020. [Online]. Available: https://tools.ietf.org/html/rfc4122

[32] *Host-Based Card Emulation Overview*. Android Developers. Accessed: Aug. 2, 2021. [Online]. Available: https://developer.android.com/guide/topics/connectivity/nfc/hce?hl=pl

[33] M. Burrows, M. Abadi, and R. Needham, ''A logic of authentication,'' *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, Feb. 1990, doi: 10.1145/77648.77649.

[34] M. Kurkowski and M. Srebrny, ''A quantifier-free first-order knowledge logic of authentication,'' *Fundam. Inf.*, vol. 72, nos. 1–3, pp. 263–282, Jan. 2006.

[35] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P. C. Héam, O. Kouchnarenko, J. Mantovani, and S. Mödersheim, ''The AVISPA tool for the automated validation of internet security protocols and applications,'' in *Computer Aided Verification*. Berlin, Germany: Springer, 2005, pp. 281–285, doi: 10.1007/11513988_27.

[36] D. Basin, C. Cremers, and C. Meadows, ''Model checking security protocols,'' in *Handbook Model Checking*, E. M. Clarke, T. A. Henzinger, H. Veith, R. Bloem, Eds. Cham, Switzerland: Springer, 2018, pp. 727–762, doi: 10.1007/978-3-319-10575-8_22.

[37] D. Basin, C. Cremers, J. Dreier, and R. Sasse, ''Symbolically analyzing security protocols using tamarin,'' *ACM SIGLOG*, vol. 4, no. 4, pp. 19–30, Nov. 2017, doi: 10.1145/3157831.3157835.

[38] M. Kurkowski and W. Penczek, ''Applying timed automata to model checking of security protocols,'' in *Handbook of Finite State Based Models and Applications*. Boca Raton, FL, USA: CRC Press, 2012.

[39] O. Siedlecka-Lamch, S. Szymoniak, and M. Kurkowski, ''A fast method for security protocols verification,'' in *Computer Information Systems and Industrial Management*. Cham, Switzerland: Springer, 2019, pp. 523–534, doi: 10.1007/978-3-030-28957-7_43.

[40] A. V. Hess and S. Modersheim, ''Formalizing and proving a typing result for security protocols in Isabelle/HOL,'' in *Proc. IEEE 30th Comput. Secur. Found. Symp. (CSF)*, Aug. 2017, pp. 451–463, doi: 10.1109/CSF.2017.27.

[41] A. Armando and L. Compagna, ''SATMC: A SAT-based model checker for security protocols,'' in *Logics in Artificial Intelligence*. Berlin, Germany: Springer, 2004, pp. 730–733, doi: 10.1007/978-3-540-30227-8_68.

[42] L. Li, J. Sun, Y. Liu, M. Sun, and J.-S. Dong, "A formal specification and verification framework for timed security protocols," *IEEE Trans. Softw. Eng.*, vol. 44, no. 8, pp. 725–746, Aug. 2018, doi: 10.1109/TSE.2017.2712621.

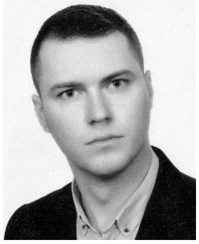
[43] M. Benerecetti, N. Cuomo, and A. Peron, "TPMC: A model checker for Time–Sensitive security protocols," *J. Comput.*, vol. 4, no. 5, pp. 366–377, May 2009, doi: 10.4304/jcp.4.5.366-377.

[44] A. M. Zbrzezny, A. Zbrzezny, S. Szymoniak, O. Siedlecka-Lamch, and M. Kurkowski, "VerSecTis—An agent based model checker for security protocols," in *Proc. 19th Int. Conf. Auto. Agents MultiAgent Syst.*, Richland, SC, USA, May 2020, pp. 2123–2125.

[45] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983, doi: 10.1109/TIT.1983.1056650.

[46] A. Grosser, M. Kurkowski, J. Piątkowski, and S. Szymoniak, "ProToc—An universal language for security protocols specifications," *Advances in Intelligent Systems and Computing*, vol. 342. Cham, Switzerland: Springer, Mar. 2015, pp. 237–248, doi: 10.1007/978-3-319-15147-2_20.

[47] *KeyProtection*. Android Developers. Accessed: Aug. 13, 2021. [Online]. Available: https://developer.android.com/reference/android/security/keystore/KeyProtection?hl=pl

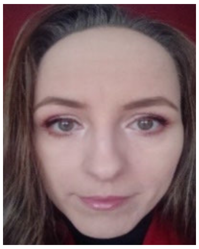


MACIEJ BARTŁOMIEJCZYK received the M.Sc. degree in software engineering from the West Pomeranian University of Technology (ZUT), Szczecin, Poland, where he is currently pursuing the Ph.D. degree. He participates in the design and implementation of transactional banking systems. He studies the design of authentication protocols, especially in a mobile environment.

IMED EL FRAY received the M.Sc. degree in naval automation from the Szczecin University of Technology, the Ph.D. degree from the Faculty of Maritime Technology, and the D.Sc. degree from the Faculty of Computer Science. He is currently a Professor with the West Pomeranian University of Technology (ZUT), Szczecin, Poland. He is the author and coauthor of four book chapters and over 80 research papers on risk assessment and risk management, information systems audit, security information and event management, and common criteria.

MIROSŁAW KURKOWSKI received the M.Sc. degree in mathematics from the Jan Dlugosz University of Czestochowa, the Ph.D. degree in mathematics and computer science from the Institute of Computer Science, Polish Academy of Sciences, and the D.Sc. degree in computer science from the Faculty of Mechanical Engineering and Computer Science, Technical University of Czestochowa. He is currently a Professor with Cardinal Stefan Wyszynski University, Warsaw, Poland. He is the author and coauthor of four books and over 60 research papers on applied mathematical logic and formal methods for the verification of software systems, especially security protocols and cryptography.

SABINA SZYMONIAK received the M.Sc. degree in computer science from the Czestochowa University of Technology, Czestochowa, Poland, and the Ph.D. degree in computer science from the Faculty of Mechanical Engineering and Computer Science, Czestochowa University of Technology. She is currently an Assistant Professor with the Czestochowa University of Technology. She is the author or coauthor of 20 research articles on the verification of security protocols and cryptography.



OLGA SIEDLECKA-LAMCH received the M.Sc. degree in engineering and the Ph.D. degree in computer science from the Czestochowa University of Technology. She was a Scholarship Holder at the Polish Academy of Sciences. Currently, she is an Assistant Professor with the Czestochowa University of Technology. She is the author or coauthor of over 30 research articles and published on formal modeling and theory of automata.