Mariusz ZIEJA

Air Force Institute of Technology
e-mail: mariusz.zieja@itwl.pl; ORCID: 0000-0003-1494-4099

Karol KAWKA

Polish Air Force University
e-mail: k.kawka@law.mil.pl; ORCID: 0000-0001-6867-0957

Konrad WOJTOWICZ

Military University of Technology
e-mail: konrad.wojtowicz@wat.edu.pl; ORCID: 0000-0001-6185-5465

Adam WETOSZKA

Polish Air Force University
e-mail: a.wetoszka@law.mil.pl; ORCID: 0000-0002-9400-3469

Tomasz PIETRZAK

National Aircraft Accidents Investigation Commission
e-mail: tomasz.pietrzak@pkbwl.gov.pl

DOI: 10.55676/asi.v4i2.70

PORTABLE BIOMETRIC MODULE SOFTWARE FOR MILITARY AVIATION SUPPORT SYSTEM

OPROGRAMOWANIE PRZENOŚNEGO MODUŁU BIOMETRYCZNEGO DO WOJSKOWEGO SYSTEMU WSPARCIA LOTNICTWA

Abstract

The article discusses a software solution that addresses the integration of portable devices into the maintenance system of the Polish Air Force. In order to meet the requirements of this solution, it is essential to have a mobile device that can collaborate with military systems on a civilian network and process classified information. The current regulations and existing solutions suggest the need to establish a procedure for wireless data synchronization through the global Internet. The scope of this solution includes developing a method for exchanging data and ensuring data security. To enhance the project, we have also implemented a streamlined user authentication process. To achieve this, we tested the biometric sensors of the mobile device, which allowed for quick and convenient user verification without compromising the security of the processed information. In the paper, we have presented the design process of this solution, along with the final results. Furthermore, we have provided an example of its implementation as a mobile application designed for the

Streszczenie

Artykuł jest odpowiedzią na zaistniałą potrzebę wykorzystania w strukturach Lotnictwa Sił Zbrojnych RP technologii mobilnych. Podstawowym wymaganiem stawianym tego typu rozwiązaniu jest wykorzystanie urządzenia przenośnego współpracującego, w oparciu o ogólnie dostępną infrastrukturę cywilną, z wojskowymi systemami działającymi w ramach sieci przetwarzającej informacje niejawne. Obowiązujące przepisy oraz aktualnie stosowane rozwiązania sugerują stworzenie procedury umożliwiającej bezprzewodową synchronizację danych za pomocą globalnej sieci Internet. Rozwiązanie swoim zakresem obejmuje opracowanie sposobu wymiany danych, razem z mechanizmem zabezpieczania danych. Dodatkowo projekt został wzbogacony o metodę skróconego uwierzytelniania użytkownika. W tym celu wykorzystane zostały czujniki biometryczne urządzenia mobilnego tj. skaner linii papilarnych. Umożliwia on szybką i komfortową weryfikację użytkownika przy zachowaniu odpowiedniego poziomu bezpieczeństwa przetwarzanych informacji. Poniżej scharakteryzowano

Artykuł wpłynął do redakcji: 12.09.2023 / Zatwierdzono do publikacji: 30.12.2023

pilot's portable device, which seamlessly integrates with the TURAWA flight safety analysis and assessment system.

Keywords: mobile application, air force IT system, aircraft maintenance management system, CMMS, military IT system

proces projektowania takiego rozwiązania oraz zaprezentowano otrzymane rezultaty. Jako przykład wykorzystano dedykowaną dla pilota aplikację mobilną współpracującą z systemem analizy i oceny bezpieczeństwa lotów TURAWA.

Słowa kluczowe: aplikacja mobilna, komputerowy system zarządzania eksploatacją, system wsparcia lotnictwa, system zarządzania eksploatacją statków powietrznych, wojskowy system informatyczny

1. INTRODUCTION

We note a significant increase in mobile hardware and software applications in database systems and solutions. Every bank, mobile network, and most online stores have introduced their mobile application. The application presented in this paper is a part of the system connected to a database that allows secure access to the collected information. This solution benefits the user by authorizing him to manage his account quickly and easily. To make their offer more attractive and, consequently, to acquire new customers, various types of institutions have decided to introduce applications for mobile devices. There is a solid upward trend in implementing this type of application[1].

The use of commonly known mobile technology is not limited to civilian applications. The best example is the usage of mobile devices in the US Air Force. Selected iOS and Android devices have access to the internal Air Force network. Mobile device manufacturers are interested in supplying their products to the military market. Therefore, Apple and Google have agreed (Cooperative Research and Development Agreement) on mutual cooperation in research to increase mobile technology security[2].

The Air Force Institute of Technology is currently working on possibly using portable devices in Polish military aviation. Aviation support systems such as "Turawa" and "Samanta" operate in the military network "MILNET-Z". Direct synchronization with the central system via wireless transmission is not possible. The article presents a solution enabling indirect data synchronization between two separate systems.

The presented solution was developed based on a mobile application that enables access to data collected in the flight safety assessment and analysis system - "Turawa." This program is intended to help the pilot of a military aircraft supervise and implement preventive and training projects to eliminate or reduce phenomena that negatively impact flight safety. This software should ensure wireless data synchronization via the global Internet.

[1] Lezcano I., Romero J., Gonzalez L., Zacarías F., Dominguez M., Centurión C., Mobile Applications And Their Importance In The Commercial World. Revista Gênero e Interdisciplinaridade. 4, pp. 797–811(2023), DOI: 10.51249/gei.v4i05.1705.

[2] Welsh W., Air Force begins first wave of mobile device distribution, https://www.defenseone.com/defense-systems/2013/02/air-force-begins-first-wave-of-mobile-device-distribution/191091/ [access: 1.06.2023].

2. THE IDEA OF COOPERATION WITH A SYSTEM PROCESSING SENSITIVE INFORMATION

Turawa is a system that processes sensitive information classified as "Restricted" following applicable internal regulations. Such a system should operate within the MILNET-Z military network. However, not all data stored in the system is sensitive information. Separating this data and extracting only public information from the system is possible.

The concept of implementing the Turawa_mobile subsystem is described below. It will operate independently of the central system and work based on the global Internet. During regular operation, both systems will be separated from each other (Fig. 1). This subsystem will only process explicit information that will be synchronized with the central system (Fig. 2). For effective use of the application, synchronization with the main server is required once every 24 hours. During synchronization, database servers will be disconnected from their home networks and interconnected for data exchange.

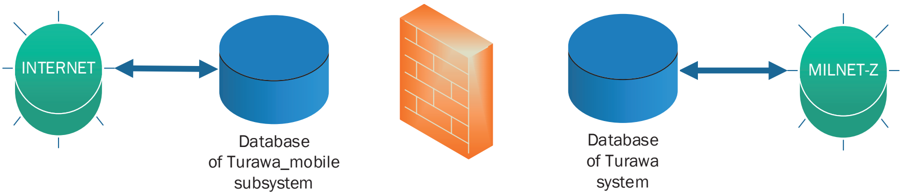

INTERNET
Database of Turawa_mobile subsystem
Database of Turawa system
MILNET-Z

Fig. 1. Scheme of expansion of the Turawa system

Source: own study.

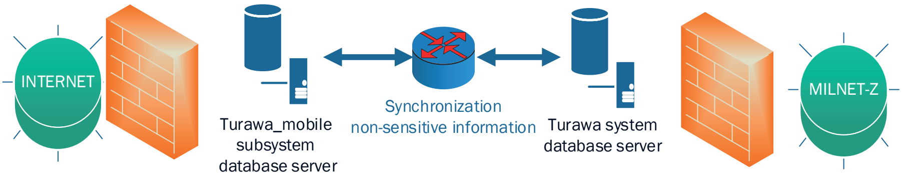

INTERNET
Turawa_mobile subsystem database server
Synchronization non-sensitive information
Turawa system database server
MILNET-Z

Fig. 2. Data synchronization scheme

Source: own study.

The data processed in the Turawa_mobile subsystem are related to a single soldier, but there will be no information enabling the physical identification of this soldier, i.e., his data. They will only be included in the central database. Each user will receive their identification key (USER_ID), thanks to which they can log in to the application. Based on this key, data binding will be performed during synchronization. The key from the central system, i.e., the user's PESEL number, will be assigned to the identification key of the Turawa mobile subsystem database. The pattern of linking the keys of two databases (Fig. 3) used during synchronization will be in the central system database.

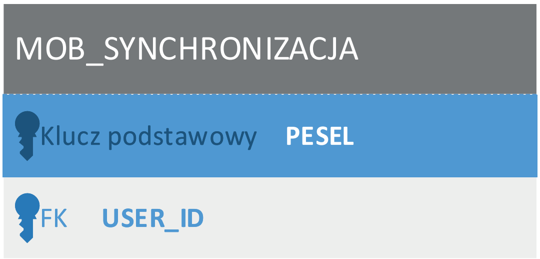


Fig. 3. Table structure MOB_SYNCHRONIZATION – database key association pattern
Source: own study.

3. INTERNET SERVICE PROJECT

The following chapter describes the program intermediating the exchange of information between the database server and the mobile application. It contains the characteristics and justification for the use of such a solution. It describes a given solution's construction, structure, and principle of operation.

3.1. USAGE OF THE INTERNET SERVICE

The mobile application should enable connection to the database via the Internet. This network may consist of many subnets, access points that operate on different principles, intermediary servers (Proxies), or various types of firewalls. However, they are considered to be the entire global Internet network. For a wide range of users' connection stability, it is necessary to use the so-called Web services, i.e., an application without a graphical user interface. This service will be an intermediary in data exchange between the mobile application and the database server (Fig. 4)[3].

[3] Redavid D., Ferilli S., Semantic Web Services Ingestion in a Process Mining Framework, Electronics (Switzerland), 12 (23), art. no. 4767 (2023), DOI: 10.3390/electronics12234767.

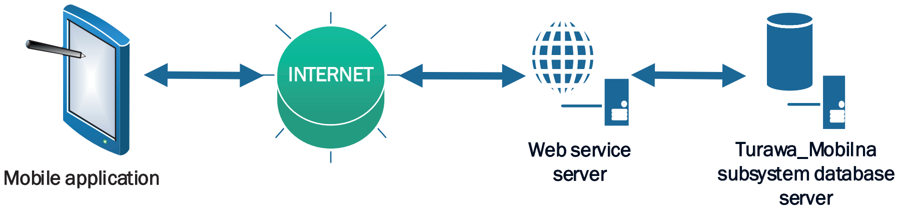


Fig. 4. Connection with the Turawa_mobilna subsystem – diagram

Source: own study.

Advantages of using the Web service:

- stateless communication (server does not create sessions = easy scalability);
- short-term connections, can quickly return to the previous place and resume the connection;
- by using the HTTP protocol as a transport channel for Internet services, we get rid of the problem of the presence of firewalls and intermediary servers (Proxy) in the network;
- Internet service allows for the interoperability of programs written in different languages and operating on various platforms;
- the natural division of the data access layer (model) into modules, which facilitates subsequent application development;
- changes in the service implementation do not require changes in the client application;
- the database server maintains a connection (creates a session) with only one client – the Internet service, thanks to which it performs operations on data faster;
- accessing the database directly from client devices is impossible, significantly impacting security.

3.2. APPLICATION PROGRAMMING INTERFACE – REST API

API (Application Programming Interface) is how programs communicate with each other. It is understood as a strictly defined set of rules and their descriptions. In other words, API allows you to use functionalities another program provides in your application. In practice, it is usually a library offering methods that enable the implementation of specific tasks. This solution is prevalent in network communication, where one server provides data for various applications[4].

REST (Representational State Transfer) is a software architecture pattern defining how to formulate API queries. In other words, it is a set of rules that the programmer should follow. It facilitates handling requests and responses, so there is no need to

[4] Lin Q., Lin B., Zhang D., Wu J., Chen X., HMS-REST v1.0: A plugin for the HEC-HMS model to provide RESTful services. Environmental Modelling and Software, 170, art. no. 105860 (2023), DOI: 10.1016/j.envsoft.2023.105860.

refer to complex documentation. Recently, it has become the leading standard for network architecture, mainly due to its simplicity and reliability[5].

The REST architecture is beneficial in situations where link bandwidth is limited, and it plays an important role. It is crucial for devices with performance barriers, e.g., palmtops, smartphones, and tablets.

The following principles characterize the Rest model:

- URL identifies all resources;
- resources can have multiple representations;
- standard HTTP methods are used to download, edit, and delete resources;
- the server does not store state information.

Characteristics of selected HTTP methods used to support the REST-API interface:

- GET – downloading information about the indicated resource (e.g., all data from the table);
- POST – an order to create a new resource with the given parameters (e.g., a new record in the table);
- PUT – specifying the changes that should be made to a given entity (e.g., editing a single data record);
- DELETE – removing resources from the server.

3.3. PROGRAM STRUCTURE

With the new version of JDeveloper 12.2.1.2, this environment enables the generation of a new type of application, the so-called ADF REST Web Application. By default, such an application includes two projects:

- RESTModel – representing the data layer. It is where business components should be implemented. They become available to the application view after declaring them in the module settings as REST WebService resources and assigning them an identifier (URI).
- RESTWebService – representing the application view, implementing HTTP query support using the ADF framework to run the application. It is generated automatically, does not require any user intervention, and supports REST components of the model resources by default.

As mentioned above, the solution enables using business components as resources for a REST service, significantly reducing the need to write program code using model layer components (XML files representing entities/data are generated), which can be easily implemented in the application.

[5] https://docs.oracle.com/middleware/12213/adf/develop/creating-adf-restful-web-services-application-modules.htm [access: 6.11.2023].

Types of business components:

- Entity objects;
- Views;
- Read-only views;
- Application modules.

Entity Objects represent the business model, data, and data validations. Each object corresponds to one table. It is possible to link entity objects using associations. By default, they consist of an Emp.xml file that characterizes a given component and EmpImpl.java representing individual lines and containing setter/getter methods for attributes.

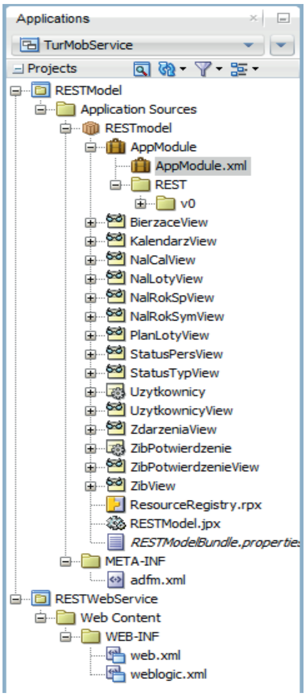
View Objects make data available to applications and can be adapted to the application's requirements by selecting attributes, adding WHERE and ORDER BY clauses, adding enumeration attributes, and relying on several entity objects. By default, they consist of the EmpView.xml file, which characterizes a given component, and EmpViewImpl.java, representing the view and containing business methods that perform operations on the data.

Read-only views (Entity-less View Objects) are a particular type of view that is not based on entities but on an explicit query in SQL. If the view is not used to modify data, omitting the entity and basing it on an SQL query increases performance. By default, it consists of an EmpView.xml file characterizing the component and containing a query in SQL.

Application Modules - their main task is to publish views for applications. The application connects directly to the module, redirecting it to a specific view. It is a set of entities and views used to perform a specific task; the smaller the module, the greater its effectiveness. Modules can be nested, where the parent module is responsible for the operation. By default, they consist of an AppModule.xml file characterizing a given component and, optionally, AppModuleImp.java containing added business methods.

3.4. BUILDING A WEB SERVICE

The web service was created using the Oracle ADF framework in the JDeveloper environment. The application model (the part representing the data layer) is based on the ORACLE 10g database. Connection to the database is achieved using the JDBC driver, which allows database connection from an application written in Java. The structure of the application is presented in Figure 5.



```
Applications
TurMobService
Projects
RESTModel
  Application Sources
    RESTmodel
      AppModule
        AppModule.xml
        REST
          v0
      BierzaceView
      KalendarzView
      NalCalView
      NalLotyView
      NalRokSpView
      NalRokSymView
      PlanLotyView
      StatusPersView
      StatusTypView
      Uzytkownicy
      UzytkownicyView
      ZdarzeniaView
      ZibPotwierdzenie
      ZibPotwierdzenieView
      ZibView
      ResourceRegistry.rpx
      RESTModel.jpx
      RESTModelBundle.propertie
  META-INF
    adfm.xml
RESTWebService
  Web Content
    WEB-INF
      web.xml
      weblogic.xml
```

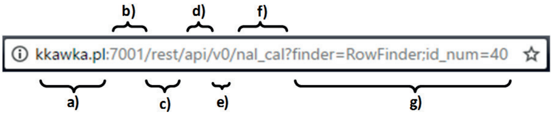
Fig. 5. Web service components

Source: own study.

Application resources, i.e., all Views to be directly available from the application level, must be declared in the application module settings and their kind and type defined. It involves assigning an access identifier (part of the URL) to each View.

The access path to a specific resource is described by the following parameters (Fig. 6):

- server IP address/domain name;
- access port number;
- application name (defined in the RESTWebService project settings - web.xml file);
- access pattern (element required by the ADF framework);
- application version;
- resource access identifier;
- optionally – input parameters of an SQL query or a data filtering query.



```
kkawka.pl:7001/rest/api/v0/nal_cal?finder=RowFinder;id_num=40
```

Fig. 6. Service access path – component parameters

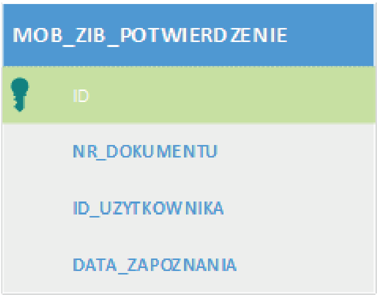
Source: own study.

3.5. DATA EXCHANGE

The web service usually retrieves and shares information from a database, mainly consisting of read-only View Objects. Each object of this type contains PL/SQL query declarations and represents a specific information resource. It is possible to enter a parameter into such a query and send it along with the URL. This query may be a complex function, and the generated information may result from many operations. Such a query often references functions and procedures performed directly on the database server.

Uploading data into the system is slightly more complex, and declarations such as PUT, INSERT, or UPDATE need to be used, referring to Entity Objects. One example of such a solution is described below, where the operation of reading the document may be confirmed by generating a "ZibConfirmation" object corresponding to this situation.

The "ZibConfirmation" object is an image of a table (Fig. 7) consisting of four fields containing records allowing the identification of a specific document and the user and then entering the date of reading the document.



MOB_ZIB_POTWIERDZENIE

ID

NR_DOKUMENTU

ID_UZYTKOWNIKA

DATA_ZAPOZNANIA

Fig. 7. Table structure "MOB_ZIB_CONFIRMATION"

Source: own study.

Operations can be performed in the above table using standard X-HTTP methods, such as GET, POST, PUT, PATCH, and DELETE. When updating the confirmation of reading the document, it looks like this:

```
URI: http://kkawka.pl:7001/rest/api/v0/zib_potwierdzenie
Content-Type: application/vnd.oracle.adf.resourceitem+json
X-HTTP-Method-Override: POST
Body:
{
  "NrDokumentu": "8732/D4/1/16",
  "IdUzytkownika": "56658",
  "DataZapoznania": "2017-05-12T02:00:00+02:00"
  }
```

The Internet service interprets the request and adds a new record on the MOB_ZIB_CONFIRMATION object (Fig. 7).

4. MOBILE APPLICATION DESIGN

The application for Android devices was built in the AndroidStudio environment, using the JAVA object-oriented programming language and the XML markup language. The technological solutions developed in the project are characterized below.

4.1. USER AUTHENTICATION

Authentication is understood as the process of confirming the declared identity. The main task of the authentication process is to obtain the required level of confidence that a given entity is aware of who it claims to be. If the user enters his ID and password, it can be assumed that they are known only to him, and by providing them for verification, he confirms that he is the owner of a specific identity.

The length of the password and the variety of characters used directly impact its security. The more characters used in a password, the more attempts an attacker must take to crack the password during a brute-force attack.

Therefore, entering a long, complicated password to access the application looks reasonable. Such a password would consist of a minimum of 12 characters, upper and lower case letters, numbers, and special symbols. It significantly increases the level of security, but entering such a complicated password every time can be pretty tedious.

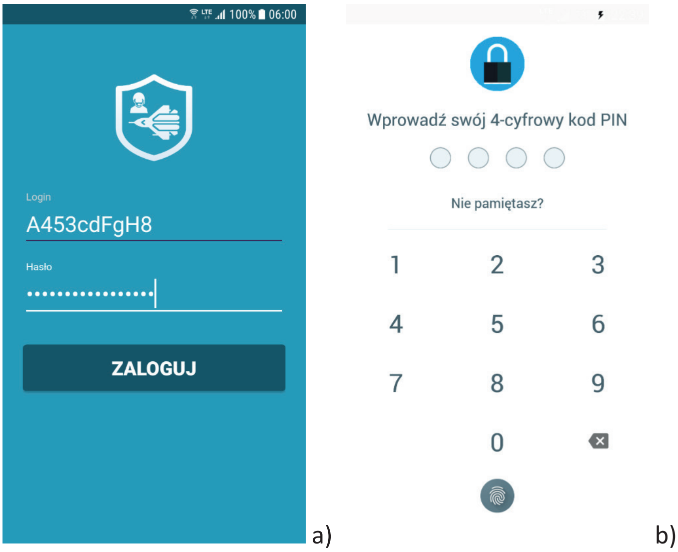
4.1.1. SHORTENED LOGIN

In recent years, the idea that a PIN code is more secure than a standard password lock has become increasingly popular. The best example is the new function Hello PIN, which was introduced in the Windows 10 operating system. Therefore, when designing a mobile application, it was decided to implement a similar solution.

The pin is assigned to the device on which it was set. If someone captures your account's master password, they can log in from multiple devices, but if they only obtain the PIN, they will gain nothing until they receive the physical device.

In the discussed project, neither the PIN code nor the password are physically saved in the application for security reasons. Their hashes are made using the SHA 1 hash function and stored in the SharedPreferences memory. The PIN code hash enables verification of the correctness of the entered code, and the password hash is sent to the server to confirm the user's identity (More about hash functions in section 4.2).

DOI: 10.55676/asi.v4i2.70



Login

A453cdFgH8

Hasło

••••••••••••••••

Wprowadź swój 4-cyfrowy kod PIN

Nie pamiętasz?

Fig. 8. User authentication – screens: a) login; b) shortened login

Źródło: own study.

When logging in to the application for the first time, you must enter the entire password (Fig. 8a), and access verification will be carried out in a shortened form (Fig. 8b). Of course, the user can choose in the application settings whether he wants to use the shortened login option. In addition to the PIN code, it is possible to log in using biometric security, i.e., fingerprint. However, this option depends on your device, which must be equipped with a fingerprint reader and appropriate software.

4.1.2. BIOMETRIC VERIFICATION IN A MOBILE APPLICATION

Biometric security is based on biometric data, i.e., information that automatically recognizes people based on physical characteristics. Biometrics deals with identifying a person based on their characteristic features, such as the image of the iris of the eye, the arrangement of fingerprints, the geometry of the voice, the shape of the face, as well as behavioral features (e.g., the way of walking, a handwritten signature). The examined features are unique to each person, thanks to which we can identify the person.

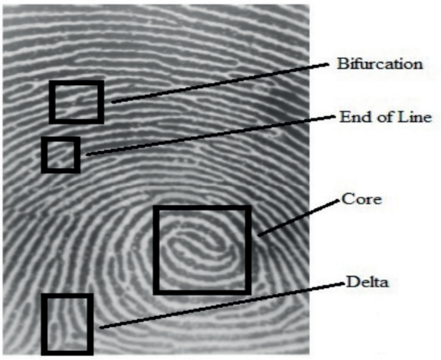


Fig. 9. Fingerprint characteristics

Source: Luna-Ortega C., Ramírez-Márquez J., Mora-González M., Romo M. C. J., López-Luévano C., Fingerprint Verification Using the Center of Mass and Learning Vector Quantization. Proceedings - 2013 12th Mexican International Conference on Artificial Intelligence, MICAI 2013, DOI: 10.1109/MICAI.2013.21.

Fingerprint readers do not analyze the entire fingerprint system, but several characteristic elements are called minutiae. These are specific features such as bifurcation, end of the line, core, and delta (. The arrangement of minutiae on the fingertip identifies an individual. To verify the consistency of the prints, a dozen or so standard features are enough, such as, for example, a specific minutiae lying in the same place. Most authorization mechanisms based on a fingerprint scanner approve the measurement when the number of minutiae matches approximately 40% of the stored fingerprint. It is the optimal value, sufficient to identify the user and thus make the scanner error-resistant. In the case of higher requirements, repeating the measurement is often necessary[6].

In February 2014, Samsung was the first company (in the Galaxy S5 model) to enable developers to use the fingerprint scanner in their applications. It was a different approach to its largest market competitor, Apple, whose software did not allow third parties to access this resource. Apple used the fingerprint scanner only in its applications, while Samsung wanted developers to implement this solution to secure their applications widely.

In 2015, the creators of the Android system, wanting to meet development trends and seeing that more and more manufacturers were implementing a fingerprint scanner, decided to unify this solution and introduce one standard API. With the release of Android 6 Marshmallow, developers received extensive documentation on implementing a fingerprint scanner in their applications. This solution enabled the rapid spread of biometric security in mobile applications as the applications became universal. Regardless of the manufacturer, all new devices running Android 6 and higher operate based on the same application source code.

[6] Luna-Ortega C., Ramírez-Márquez J., Mora-González M., Romo M. C. J., López-Luévano C., Fingerprint Verification Using the Center of Mass and Learning Vector Quantization. Proceedings - 2013 12th Mexican International Conference on Artificial Intelligence, MICAI 2013, DOI: 10.1109/MICAI.2013.21..

Fingerprints are only available on a specific device and are not shared with Google or other applications. The application is only notified about fingerprint verification. The developers of Android have formulated requirements for device manufacturers defining how fingerprints are stored on the device:

1. Fingerprint registration and recognition should be performed in a trusted device execution environment (secure memory area of the main processor).
2. All fingerprint data must be saved to a trusted memory or biometric reader, making fingerprint images inaccessible.
3. Only the encrypted version of the fingerprint can be stored in the system, even if the file system itself is also encrypted.
4. Deleting a user must result in deleting their fingerprints.
5. If root access (an account with complete control within the operating system) is gained on the device, fingerprints must remain inaccessible.

4.2. COMMUNICATION WITH THE WEB SERVICE

As mentioned in Chapter 2, queries to the web service are made using a URI, and the response with data is returned in the JSON format. The data provided by the service, before being applied, should be converted into Java objects (POJO – Plain Old Java Object). Similarly, the data must be converted to JSON format before sending.

Retrofit[7] is a library designed to help developers communicate with REST API servers. Java annotations enable easy and quick handling of queries to the Internet service, among others. The programmer's task is limited to writing an interface containing the application's methods to call on the API and adding annotations describing which URLs each query should refer to, what parameters to send, and what data to receive. Retrofit carries out the rest of the work. The GSON converter included in the library enables the automatic mapping of Java objects. Since version 2.0, Retrofit has worked closely with the OkHttp library.

OkHttp[8] is a powerful HTTP client for Android applications. It is a library that implements reconnected connections, transparent data compression, and response buffering to avoid repeated requests. It can recover data in connection problems, supporting several API access paths (in the case of multiple servers). It contains mechanisms that significantly speed up the generation of asynchronous requests to the API and enable the use of processor multithreading. This way, data can be downloaded "in the background" while performing other activities.

Android is one of many operating systems that support multithreading. This solution allows several tasks to be divided into separate processes. To use this solution, you must specify which tasks will be performed asynchronously in the application code. All communication with the web service occurs as part of a separate process and runs "in the background" independently of the GUI.

[7] https://square.github.io/retrofit/ [access: 25.09.2022].

[8] https://square.github.io/okhttp/ [access: 21.07.2022].

5. SECURITY OF TRANSMITTED INFORMATION

Securing and protecting information is vital in rapid technological progress and ICT development. Information is a strategic resource when delivered only to the correct recipient intact. The methods of securing data are briefly described below, and the solutions used to secure wireless communication between the web service and the mobile application are presented.

The HTTPS (Hypertext Transfer Protocol Secure) protocol initiates and maintains a secure communication channel between the client and the server. Data transferred via the HTTPS application layer protocol is protected by the TLS (Transport Layer Security) presentation layer protocol.

The OpenSSL program has been used to generate your own CA and domain certificate containing libraries implementing SSL and TLS standards and cryptographic mechanisms. Additionally, it is equipped with a set of console tools for creating keys and certificates, managing a certification authority, encrypting, and calculating digital signatures[9].

5.1. CERTIFICATE PINNING

Users and developers are expecting secure, encrypted transmission. However, cryptography alone cannot provide this. An unauthorized person may obtain information enabling the decryption of transmitted data, which is very popular in the case of well-known protocols such as VPN, SSL, and TLS.

The specific conditions in which mobile devices are used force application developers to employ additional security measures. Users communicating via a wireless network often use various unverified access points, the so-called hot spots. Attackers can impersonate access points the user trusts (the so-called "Evil Twin" method). It is also possible to force the connection to a trusted point to be interrupted, after which the device resumes communication by connecting to the device of the person trying to intercept sensitive information.

The greatest threat to transmission using the TLS protocol is the so-called MITM (Man in the middle) attack, where a hidden attacker joins the communication and captures sensitive information[10]. Below a simple example (Fig. 10) demonstrating this method is presented:

1. The application connects to an open access point.
2. The Wi-Fi network is configured so that when a connection is established with a specific API (calling its www address), it is redirected to a virtual intermediary

[9] Buchanan W.J., Helme S., Woodward A., Analysis of the adoption of security headers in HTTP IET Information Security, 12 (2) (2018), pp. 118–126, DOI: 10.1049/iet-ifs.2016.0621.

[10] Elrawy M.F., Hadjidemetriou L., Laoudias C., Michael M.K., Detecting and classifying man-in-the-middle attacks in the private area network of smart grids, Sustainable Energy, Grids and Networks, 36, art. no. 101167 (2023), DOI: 10.1016/j.segan.2023.101167.

server. This server also has a forged certificate signed by a trusted Certification Authority (CA).
3. The proxy server connects to the target server, pretending to be a mobile application.
4. The intermediary server conducts encrypted transmissions with the mobile application and the target server with two asymmetric encryption keys.
5. The proxy server records all data exchange between the mobile application and the API server.

This way, all communication passes through an intermediary server (e.g., the attacker's laptop). Thus, an unauthorized person can collect much information about how the system works and even effectively impersonate authorized persons, gaining access to hidden resources.

The MITM method can be used not only when operating a wireless Wi-Fi network, but this is the simplest and most popular solution; when using other transmission media, the principle of operation is very similar.

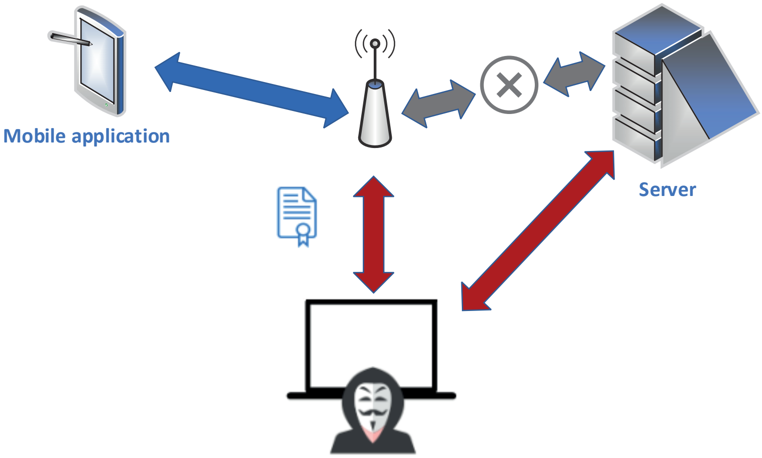


Fig. 10. Scheme of interference in data transmission – MITM method
Source: own study.

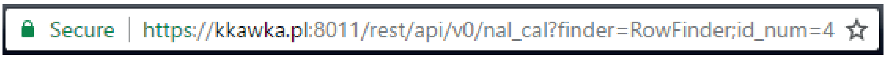
An effective authentication mechanism for the target server should be used to exclude the possibility of an MITM attack. This effect can be achieved by implementing the so-called certificate pinning. This process aims to prevent the acceptance of the so-called improvised TLS certificates used to impersonate one of the communication elements. Certificate pinning involves associating the host address with the expected certificate. Typically, certificates are verified by checking the signature hierarchy. The certificate will be automatically accepted if a CA signs it on the trusted list (defined in the operating system). One method of pinning certificates is to ignore the default list of trusted CAs and trust only a specific, user-defined CA.

To sum up, it involves verifying whether the website with which the application communicates is what it claims to be. It is done by comparing the certificate sent by the website server with the certificate saved in the application. This form of authentication effectively protects against MITM attacks and has been implemented in the designed mobile application.

5.2. TEST OF IMPLEMENTED SOLUTIONS

The application is hosted on the Oracle Weblogic 12c server. The server has been configured to support the HTTPS protocol on port number 8011. It has implemented its certificate repository.

It was necessary to install the trusted CA repository generated in the previous point of your own CA to test the connection's correctness using the Google Chrome web browser. After completing the activities mentioned above, the web browser successfully established the connection utilizing the HTTPS protocol, as evidenced by the "padlock" logo and the inscription "Secure," confirming secure data transmission (Fig. 11).



Secure | https://kkawka.pl:8011/rest/api/v0/nal_cal?finder=RowFinder;id_num=4

Fig. 11. HTTPS protocol test – using a web browser

Source: own study.

The next stage was to use the OpenSLL console environment to conduct a comprehensive connection test. The test confirmed the correct implementation of secure data transmission protocols. He presented all data encryption methods supported by the server, the type of certificate signature used, the issuer (Tur-Cert), and the intended purpose of the certificate.

6. CONCLUSIONS

The presented solution enables portable devices in military aviation support systems. It fulfills its role when full real-time data synchronization is not required for the system's proper operation. An application for supervising preventive and training activities fits perfectly into this strategy.

The possibilities of using mobile devices in military aviation do not end with 24-hour data synchronization. The solution should be intensively developed to access all data resources without time limits. However, it makes it much easier to implement mobile technologies in the Polish Air Force.

The project used modern information exchange solutions, typical of the rapidly developing mobile application sector. Data is transferred in encrypted form using secure transmission protocols. Currently, most mobile applications of financial institutions

DOI: 10.55676/asi.v4i2.70

and banks implement the same data transmission security solution. The implemented technology guarantees high security for sensitive information processed in the system.

Considering development trends worldwide, it can be concluded that the use of mobile technology in all military aviation IT support systems is a matter of the near future, as well as its use in other types of Polish Armed Forces. Many concepts and research are being carried out using mobile devices to communicate and supervise individual soldiers. One of the concerns is ensuring the security of transmitted information. It is a rapidly developing area, and many institutions are working to improve security. However, using this technology in the banking sector or the US Air Force proves that solutions that meet high-security standards are currently available. Therefore, applications of this type should be successfully used in aviation.

BIBLIOGRAPHY

Buchanan W.J., Helme S., Woodward A., Analysis of the adoption of security headers in HTTP (2018), IET Information Security, 12 (2), pp. 118–126, DOI: 10.1049/iet-ifs.2016.0621.

Elrawy M.F., Hadjidemetriou L., Laoudias C., Michael M.K., Detecting and classifying man-in-the-middle attacks in the private area network of smart grids Sustainable Energy, Grids and Networks, 36, art. no. 101167 (2023), DOI: 10.1016/j.segan.2023.101167.

https://docs.oracle.com/middleware/12213/adf/develop/creating-adf-restful-web-services-application-modules.htm [access: 6.11.2023].

https://silo.tips/download/tworzenie-aplikacji-j2ee-w-oparciu-o-oracle-application-development-framework-ad [access: 26.10.2023].

https://square.github.io/okhttp/ [access: 21.07.2022].

https://square.github.io/retrofit/ [access: 25.09.2022].

https://www.soais.com/rest-services-in-oracle-adf/ [access: 19.11.2023].

Kawka K., Wojtowicz K., Zieja M., Examination and evaluation of training jet aircraft maintainability. Proceedings 2020 IEEE International Workshop on Metrology for Aerospace, IEE, 2020, DOI: 10.1109/MetroAeroSpace48742.2020.9160188.

Kowalski M., Izdebski M., Żak J., Gołda P., Manerowski J., Planning and management of aircraft maintenance using a genetic algorithm. Eksploatacja i Niezawodnosc, 23 (1), pp. 143–153 (2021), DOI: 10.17531/EIN.2021.1.15.

Lewitowicz J., Szelmanowski A., Pazur A., Janik P., Computer-based management system for reliability and operational readiness of the integrated communication system for military helicopters. AUTOBUSY – Technika Eksploatacja Systemy Transportowe, 20, pp. 303–308 (2019), DOI: 10.24136/atest.2019.055.

Lezcano I., Romero J., Gonzalez L., Zacarías F., Dominguez M., Centurión C., Mobile Applications And Their Importance In The Commercial World. Revista Gênero e Interdisciplinaridade, 4, pp. 797–811(2023), DOI: 10.51249/gei.v4i05.1705.

Lin Q., Lin B., Zhang D., Wu J., Chen X., HMS-REST v1.0: A plugin for the HEC-HMS model to provide RESTful services. Environmental Modelling and Software, 170, art. no. 105860, (2023), DOI: 10.1016/j.envsoft.2023.105860.

Mitra S., Gofman M., Biometrics in a Data Driven World: Trends, Technologies, and Challenges, CRC Press, Boca Raton 2017.

Pigłas M., Radoń T., Szymanski M., Krutkow A., Przystawska A., Information system support for military aircraft operations SI SAMANTA as a tool to support logistic resource management. Journal of KONBiN, 50, 269–286, (2020), DOI: 10.2478/jok-2020-0086.

Reda K., Kedzierski M., Detection, classification and boundary regularization of buildings in satellite imagery using faster edge region convolutional neural networks. Remote Sensing, 12 (14), art. no. 2240, (2020), DOI: 10.3390/rs12142240.

Redavid D., Ferilli S., Semantic Web Services Ingestion in a Process Mining Framework, Electronics (Switzerland), 12 (23), art. no. 4767, (2023), DOI: 10.3390/electronics12234767.

STANAG 5066, Profile for High Frequency (HF) Radio Data Communications, Edition 4, North Atlantic Treaty Organization, 2021.

Tam-Seto L., Wood V., Linden B., Stuart H., Perceptions of an AI-Supported Mobile App for Military Health in the Canadian Armed Forces. Military Behavioral Health, 9 (2020), DOI: 10.1080/21635781.2020.1838364.

Zieja M., Smoliński H., Gołda P., Jakościowe i ilościowe szacowanie ryzyka na podstawie analizy zdarzeń w lotnictwie wojskowym. Research Works of Air Force Institute of Technology, 38 (2016), DOI: 10.1515/afit-2016-0007.

Żyluk A., Zieja M., Adamski M., Kawka K., Maintaining A Continuous Readiness For Military Pilot Flights By Using Mobile Technology. Journal of KONBiN 2019 Volume 9, Issue 4, DOI: 10.2478/jok-2019-0099.