Authentication Enhancement Using Mobile-Based Application

Ibrahim Fadul Ibrahim Osman
College of Ahfad for Women
Ahfad University
Omdurman, Sudan
bimco758@gmail.com

Yasir Abdelgadir Mohamed
Faculty of Computer Science & IT
Karary University
Omdurman, Sudan
yasir_eym@yahoo.com

***Abstract*—Information security is one of the most complicated and difficult issues to which all interested parties attach a great deal of attention. Security and confidentiality studies of information, therefore, prevail over studies in other fields. The loss of information and the destruction of it by falling into hands of the abusers have many images and different forms, all of which focus on circumventing the regulations and entering them illegally. In this paper, an application Android relies on the many advantages of the Android system to authenticate a user on the mobile phone, so that they can login to a system that built, installed on the environment of the computer. The application is designed depending on the features of the mobile operating system, in the secure transport and control of data transmission, in addition to the features of the mobile phone itself, thus, features integrated to raise the level of protection of information to increase the strength authenticate to a higher extent.**

Keywords—infosec; confidentiality; integrity; availability; authentication; access resource.

1. INTRODUCTION

Security is to define information from unauthorized access, use, disruption, disclosure, modification, recording or destruction. [1] The great technological advances, the development of various means of connection and communication, the openness of the world to each other and the reliance on the transmission of various types of data over the networks have led to a risk of leakage of these data and access to the wrong people or competitors, and thus became an urgent need to maintain information security. Information security "well-informed sense of assurance that the information risks and controls are in balance" as Jim Anderson, said. [2] Application or computer system security encompasses measures taken throughout the code's life-cycle to prevent gaps in the security policy of an application or the underlying system (vulnerabilities) through flaws in the design, development, deployment, upgrade, or maintenance or database of the application. [3] Applications only control the kind of resources granted to them, and not which resources are granted to them. They, in turn, determine the use of these resources by users of the application through application security. Authentication is relevant to multiple fields. In art, antiques and anthropology, authentication is verifying that a given artefact was produced by a certain person or in a certain place or period of history. In computer science, verifying a person's identity is often required to allow access to confidential data or systems. Authentication is the process of actually confirming that identity. It might involve confirming the identity of a person by validating their identity documents and verifying the authenticity.

2. BACKGROUND

A. Java

Java is a language and a platform originated by Sun Microsystems. Sun organized Java into three main editions: Java SE, Java EE, and Java ME, so it is a language in which developers express source code (program text). Java's syntax (rules for combining symbols into language features) is partly patterned after the C and C++ languages in order to shorten the learning curve for C / C++ developers. [4] Java platform consists of a virtual machine and an execution environment. The virtual machine is a software-based processor that presents an instruction set, and it is commonly referred to as the Java Virtual Machine (JVM). The execution environment consists of libraries for running programs and interacting with the underlying operating system (also known as the native platform). The execution environment includes a huge library of prebuilt classfiles that perform common tasks, such as math operations (trigonometry, for example) and network communications. This library is commonly referred to as the standard class library. A special Java program known as the Java compiler translates source code into object code consisting of instructions that are executed by the JVM and associated data. These instructions are known as bytecode. Figure (2.1): shows this translation process. [4]

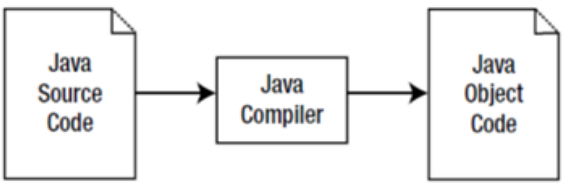

Figure (2.1): *The Java compiler translates Java source code into Java object code consisting of bytecode and associated data*

The paper takes advantage of standard Java libraries and their purpose to build an application that can be accessed through a secure mobile application because these libraries provide support for running programs and interacting with other applications at very high levels of protection.

B. Android

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

Android offers the following features: [4]

- GSM Telephony support.
- Bluetooth, EDGE, 3G, and Wi-Fi support.
- Camera, GPS, compass, and accelerometer support.
- Dalvik virtual machine optimized for mobile devices.
- Integrated browser based on the open source WebKit engine.
- Optimized graphics powered by a custom 2D graphics library; so, 3D graphics based on OpenGL ES 1.0, 1.1, 2.0, or 3.0.
- Media support for common audio, video, and image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, and GIF).
- An application framework enabling reuse and replacement of application components.

Although not part of the software stack, Android's rich development environment (including a device emulator and a plug-in for the Eclipse IDE) could also be considered an Android feature. The paper sought to find ways to exploit the advantages of the Android system to address some of the disadvantages used in traditional systems used only the password for authenticate.

C. Android Development

Contrary to might expect, Android didn't originate with Google. Instead, Android, Inc., a small Palo Alto, California-based start-up company in 2003, initially developed Android. Google bought this company in the summer of 2005 and released a beta version of the Android SDK in November 2007, a consortium of major actors in the mobile area built around Android: [5]

- Software companies: Google, eBay, etc.
- Mobile operators: T-Mobile, Telefonica, Vodafone, etc.
- Hardware vendors: Intel, Texas Instruments, Qualcomm, NVidia.
- Hardware manufacturers: HTC, Sony Ericsson, Samsung, LG, etc.

D. Android Open Source Project (AOSP)

At every new version, Google releases its source code through this project so that community and vendors can work with it. One major exception: Honey-comb has not been released because Google stated that its source code was not clean enough to release it. One can fetch the source code and contribute to it, even though the development process is very locked by Google. Only a few devices are supported through AOSP though, only the two most recent Android development phones and tablets (part of the Nexus brand) and the panda board. On September 23, 2008, Google released Android 1.0, whose core features included a web browser, camera support, Google Search, Wi-Fi and Bluetooth support, and more. This release corresponds to API Level 1. (An API level is a 1-based integer that uniquely identifies the API revision offered by an Android version; it's a way of distinguishing one significant Android release from another). [5]

E. Android Architecture

The Android software stack consists of apps at the top, a Linux kernel with various drivers at the bottom, and middleware (an application framework, libraries, and the Android runtime) in the centre. Figure (2.2): shows this layered architecture. [4]

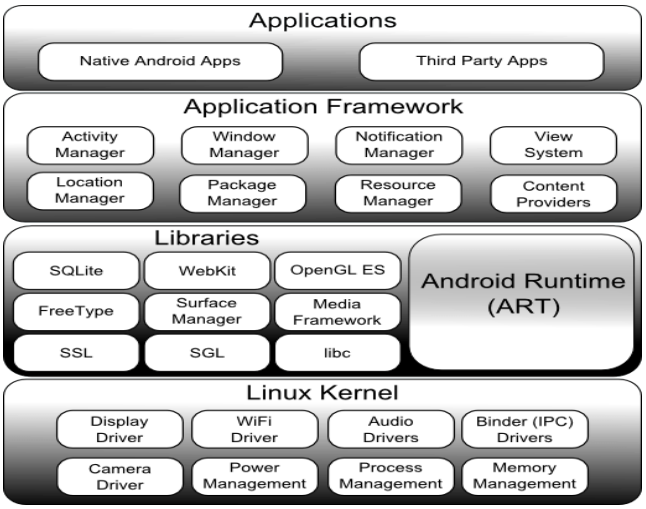

Figure (2.2): *Android's layered architecture consists of several major parts.*

F. Android Security Model

Android's architecture includes a security model that prevents apps from performing operations that are considered harmful to other apps, Linux, or users. This security model is mostly based on process level enforcement via standard Linux features (such as user and group IDs), and places processes in a security sandbox. [4]

By default, the sandbox prevents apps from reading or writing the user's private data (such as contacts or e-mails), reading or writing another app's files, performing network access, keeping the device awake, accessing the camera, and so

on. Apps that need to access the network or perform other sensitive operations must first obtain permission to do so. [4] Android handles permission requests in various ways, typically by automatically allowing or disallowing the request based upon a certificate or by prompting the user to grant or revoke the permission. Permissions required by an app are declared in the app's manifest file so that they are known to Android when the app is installed. These permissions won't subsequently change. [4]

G. Related Work

The researcher in [6] has proposed a simple, novel scheme for using a mobile device to enhance CardSpace authentication. During the process of user authentication on a PC using CardSpace, a random and short lived one-time password is sent to the user's mobile device; this must then be entered into the PC by the user when prompted. The scheme does not require any changes to login servers, the CardSpace identity selector, or to the mobile device itself.

Paper [7] suggested a secure authentication mechanism by integrating the public key with the hash-chaining technique. The propose protocol satisfies the security requirements of third generation (3G) mobile networks. Also provide the protection of the international mobile subscriber identity (IMSI) to ensure subscriber un-traceability, key refreshment periodically, strong key management and a new non-repudiation service in a simple and elegant way. The proposed protocol is more secure protocol than the other available authentication protocols. To avoid the complicated synchronization as in universal mobile telecommunications system (UMTS) the proposed protocol does not use sequence number (SEQ), the management of a hash chain is simple and elegant compared to that of SEQ. This proposed protocol is secure against network attacks, such as replay attacks, guessing attacks, and other attacks.

A network independent mobile based authentication scheme has been proposed in [8]. The framework uses the pre-shared number and MAC address of the device along with TOTP to generate a hash known as the one-time identity token to successfully authenticate the user attempting to access the services offered by the network host. Unlike SMS and location based multi-factor authentication schemes, it does not require network services to transmit or to generate the OTP for authenticating the user. It never transmits the pre-shared number and the MAC address of the device during the token generation process. MAC is only shared once through the channel at the time of application registration on token server, thus making the intruder difficult to guess and the number can also be modified by the user.

Liao in [9] has developed a slight modification on a previous work that was published by Das, Saxena [10] which presented a dynamic ID-based remote user authentication scheme using smart cards. As a result, the improved scheme has the ability to enhance the security mechanism presented by of Das and Saxena, and Gulati's scheme. In addition, the proposed scheme does not add much computational costs. Thus, Compared with Das, Saxena, and Gulati scheme, Liao and Hwang proposed, scheme is also efficient.

- Researcher in paper [6] was interested in authentication using random one-time password to the mobile device regardless of mobile user whether it is actually the user authorized to receive the message or someone else is a terrorist.

- Paper [7] proposed a partial security authentication mechanism, nevertheless it did not address other risks, such as access and processing of resources to log in in a seemingly legitimate way (guessing, injection).

- What was suggested in paper [8] is the mobile network authentication system, and, consequently, it is not safe for internal attacks that should pay adequate attention to the effects that may incurred.

- The treatments performed in the paper [10] and the improvements introduced in [9] are all efforts to improve authentication, but then again the same user status is not considered authentic

3. THE METHODOLOGY

A. Flowchart

This section discusses the application flow diagram and scenario used to ensure and secure system authentication using mobile applications. The scenario is explained as the following:

Step (1): Mobile Side Authentication

Initially the user must authenticate themselves using the mobile application by providing registered Username and Password. The application will do the following after that:

I. Verify the user, if exist; then do:

- Generate random password based on the provided password using AES algorithm.
- Concatenate the generated code with stored value called here the 'Padding', which stored previously and securely by the application and known only by the user.
- Submit the concatenated result to the server for the other side for authentication purpose.
- Show only the generated random password to the user without the Padding for confusion purpose and ensure high security.

II. Move to Step (2).

Step (2): System Side Authentication

- The secure system asks for the User ID and Authentication Password to authenticate the user.
- The user should follow the random password with the padding for successful authentication, because the User ID and Authentication Password are not enough without the Padding, which has a specific life time for high secure authentication.

B. Mobile Side Authentication

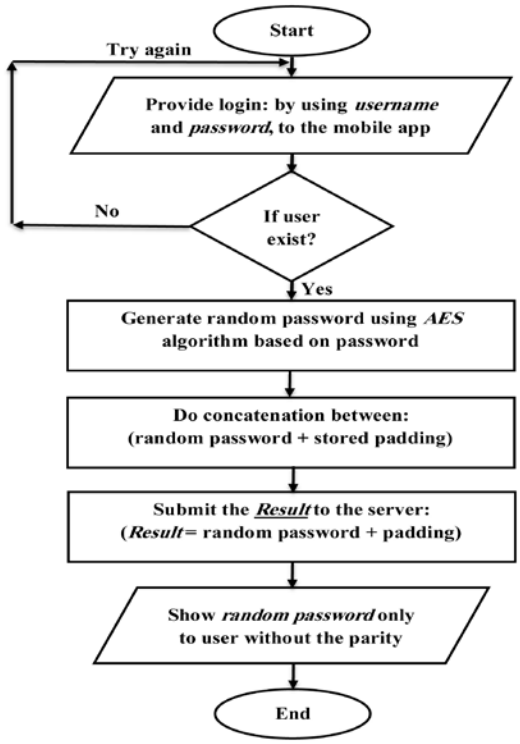

Figure (3.1): *Mobile Side Authentication.*

Step 1: Start:

The application starts with the following conditions:

- Secure connection between mobile and server.
- Achieve Steps 2.

Step 2: Login:

Provide login by using Username and Password, to the mobile app. The user must be pre-registered in the system database, and if the Username, Password do not match either or both of that information recorded in the system database, the application will not work at all.

Step 3: Generate password:

The application generates a random password using AES algorithm based on the password.

Step 4: Concatenation:

After successfully generating the random password, the application do concatenation between "password" and "padding" - the value that is pre-configured in the settings part - and then considered as a single value.

Step 5: Submit:

Submit the value that was processed in the previous step to the server.

Step 6: Show password:

In this step, the random password given by the application is displayed only without the padding added in the step 4 to make more protection level.

Step 7: End:

End steps of the mobile side authentication.

C. System Side Authentication

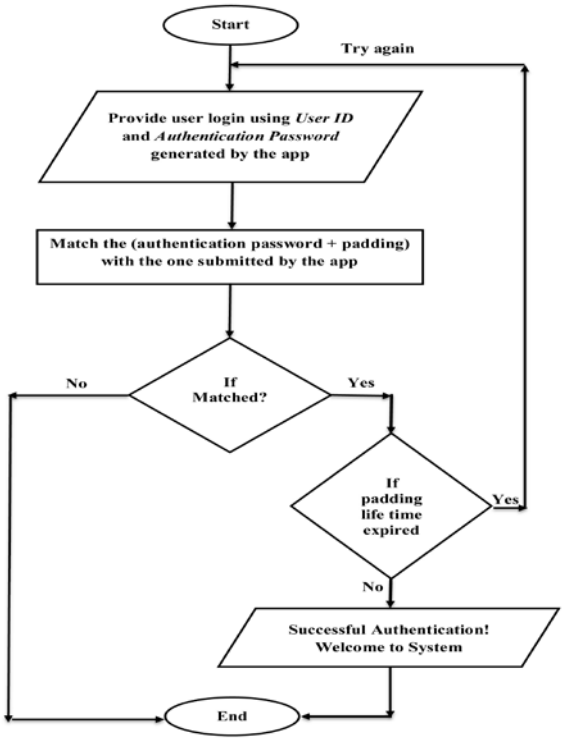

Figure (3.2): *System Side Authentication.*

Step 1: Start:

The system starts.

Step 2: Login:

Provide login by using User ID and Authentication Password generated by the app. However, you cannot take advantage of the random password generated from the mobile application alone to login to the system, the following steps must be followed:

- Add the padding of the random password generated by the mobile application.
- Achieve Steps 3.

If step 2 is not implemented, you cannot login, in this case try again all steps from the beginning.

Step 3: The matching:

The value generated by the application and sent via the mobile device to the server to be stored, as discussed in the previous section, compares them to the value entered by the system user; matching supports user authentication and mismatch requires a retry to validate authentication again, according to the specified time period.

The random password should be consumed only once in the specified period of time without exceeding 30 seconds.

Step 4: Successful Authentication:

Following all of the steps above achieves successful access to the system.

Step 5: End:

End steps of the system side authentication.

D. Sequence Diagram

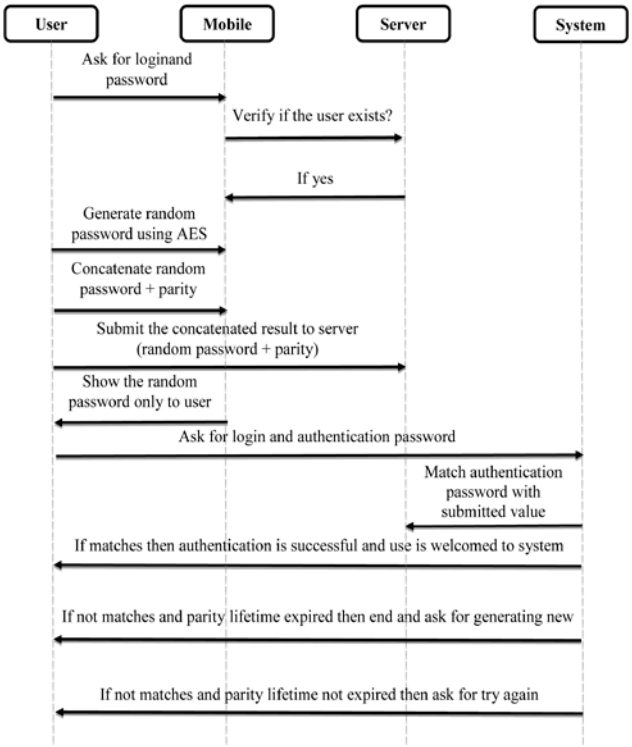

Figure (3.3): *The Sequence Diagram.*

4. DESIGN AND IMPLEMENTATION

A. User Interface for Mobile Application

The figure (4.1): Shows shortcut icon for application after installing, it appears in the distinctive ring frame within the set of applications installed on the mobile device. [11] [12]

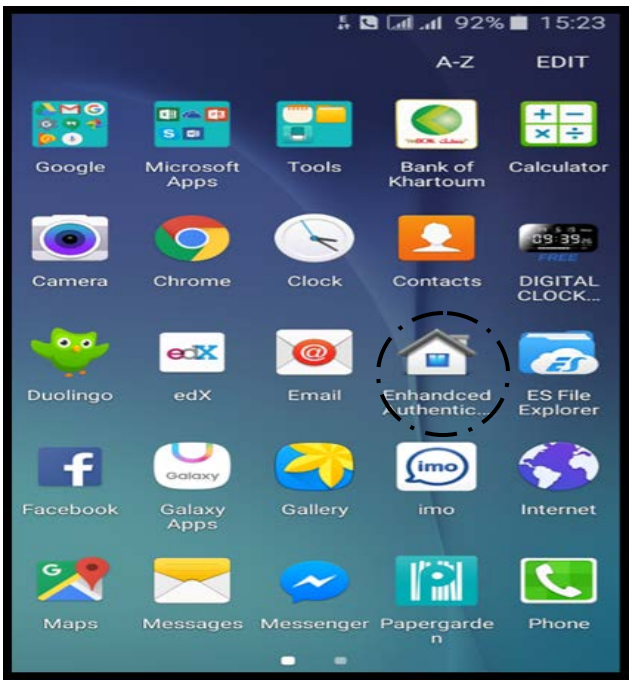

Figure (4.1): *The app icon.*

The figure (4.2): Shows the application's login screen, where the user's name and password are entered correctly, get to the next screen, unless either or both are incorrect. They must be corrected or impossibility of entry to the application.

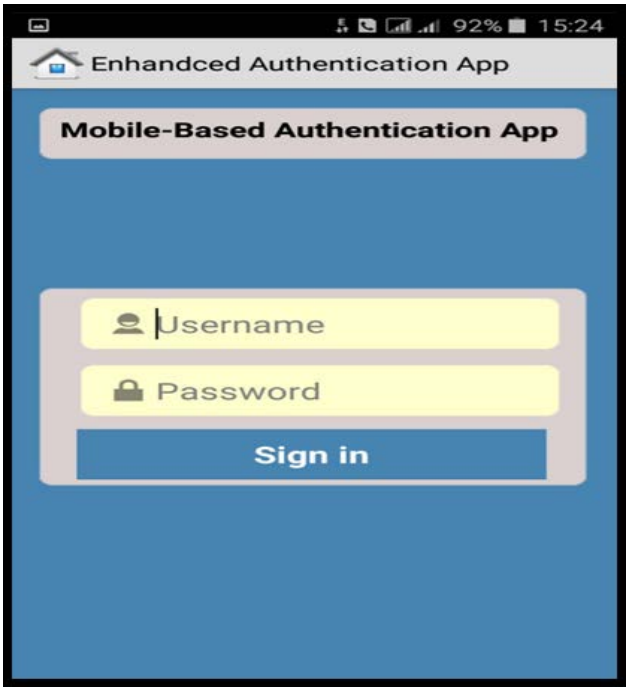

Figure (4.2): *Login screen.*

The figure (4.3): Shows the application's main screen, displays the authenticated user's name that has access to the application and generates the random password; however, any authenticated user using the application cannot use the generated random password from the application unless its padding matches the value of the owner of the application used.

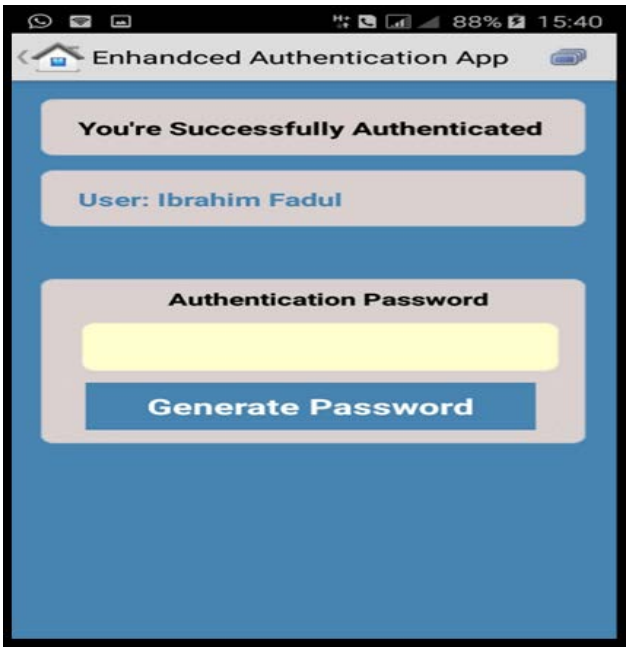

Figure (4.3): *The main screen.*

B. User Interface for System

The figure (4.4): Shows the system login screen using the user id and random password generated from the mobile application.

Figure (4.4): *The system login screen.*

The figure (4.5): Shows the information that the users table contains (id, fname, lname, username, password and pad).

id	fname	lname	username	password	pad
1200	Ibrahim	Fadul	IB	123321	NULL
1201	Hamdi	Fadul	HA	321123	NULL
1203	Abdelrahman	Fadul	AB	456654	NULL
1204	Abubaker	Abass	SA	654456	NULL
1205	Aboutalib	Mohmeed	AM	789987	NULL

Figure (4.5): *The information of users.*

C. Results of Mobile Application

The figure (4.6): Shows the application under 'application manager menu', through which the application can be controlled by uninstallation, as well as the size and characteristics of the application.

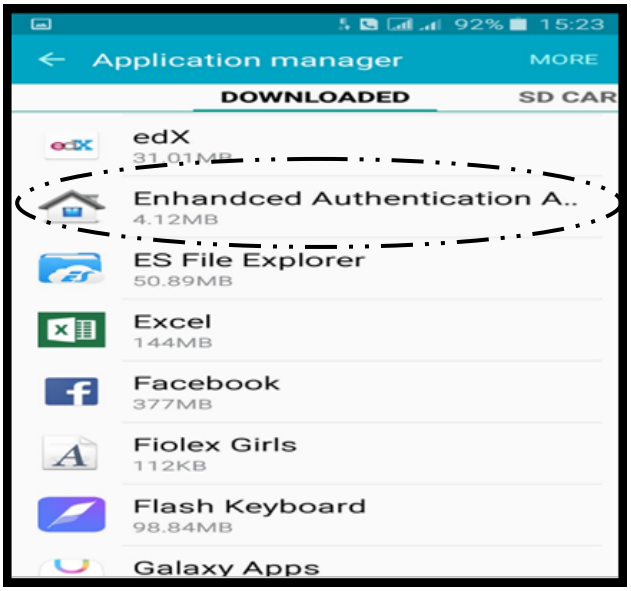

Figure (4.6): *The app in the Application manager menu.*

When the application is executed, immediately after the welcome screen, an application login screen display. This requires enter of Username, Password correctly without the slightest ambiguity.

- If no connection is available, between the mobile device and the server, the result is as shown in figure (4.7).
- If the connection is valid, then authentication of the user is done.
- If the user's name is included in the user list, and the password is correct, then the main application screen appears as in the figure (4.8) and otherwise, a message will appear indicating error as in the figure (4.10).

The figure (4.7): Shows a waiting message when the connection between the mobile device and the server is not available. The application stops if the specified time of the process exceeded.

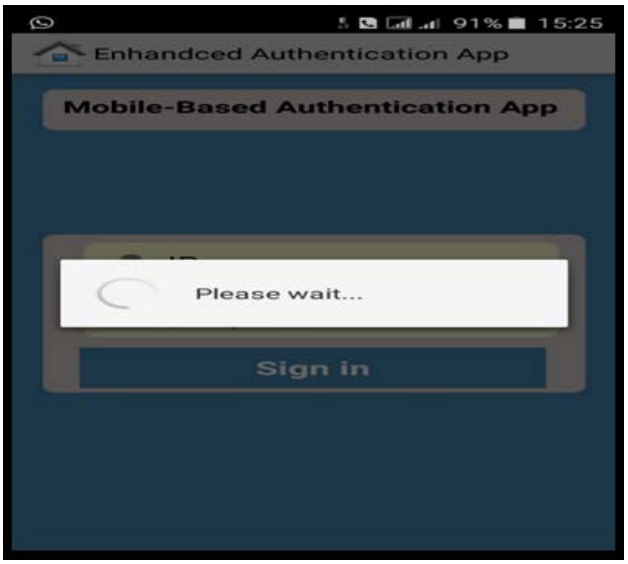

Figure 4.7: *The app at runtime mode.*

The figure (4.8): Shows the main screen, shows the authenticated user name that has access to the application and generates the random password.

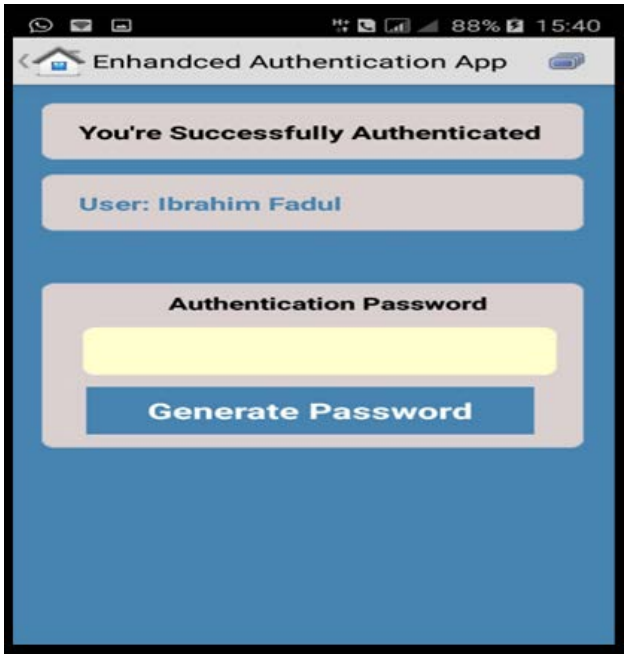

Figure (4.8): *The main screen.*

The figure (4.9): Shows the main application screen after generating a random password, and a confirmation message appears confirming that the random password generation is completed correctly. As shown by the two distinctive circles.

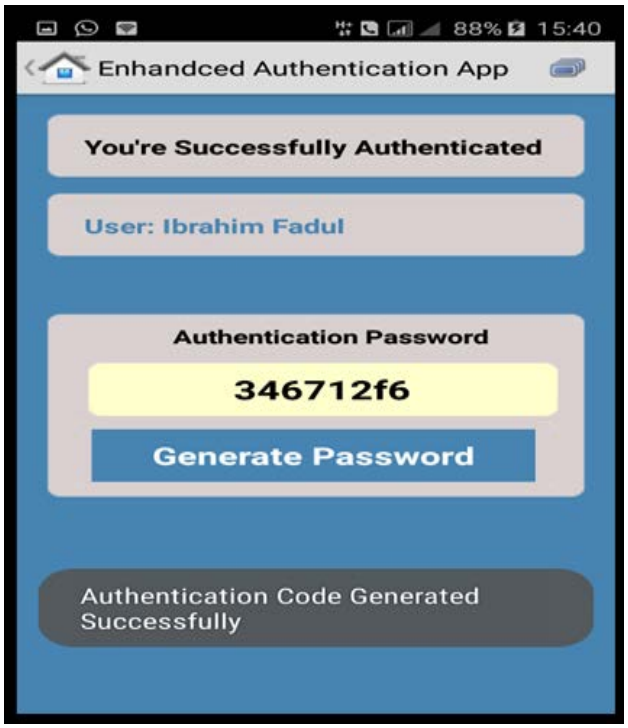

Figure (4.9): *The generate password screen.*

The figure (4.10): Shows the application login screen, if any user's name, password, or both entered incorrectly, a warning message will be generated indicating an error.

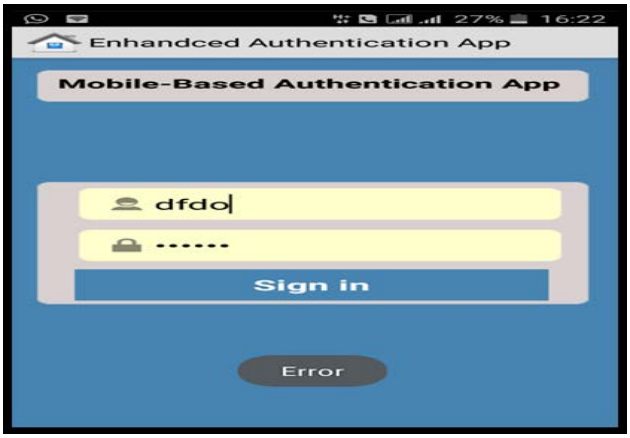

Figure (4.10): *The login screen.*

D. Results of System

After the random password generation is successfully generated, the role of the application is only of some order such as:

- Concatenate the random password with the padding and send it to the server.
- Calculation of the time after the transmission process, the random password should be used in a time period less than or equal to 30 second only.

By the other side - on the side of the system - we get the following results:

Accessing the system via the main login screen requires the User ID, Authentication Password generated by the mobile application with the padding which the user has already been setting in their own application.

- If you enter the User ID, Authentication Password by the application in addition to the padding correctly, the login of the system and access to the content becomes successful.
- If any of the above conditions are violated, the entry process, if not impossible, is very difficult, as recorded according to the most recent experiments.

The figure (4.11): Shows the login screen of the system after full verification of the system user authentication, with a notice indicating the success of the process.

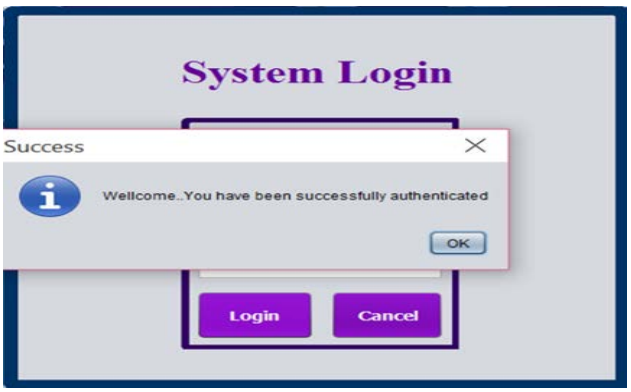

Figure (4.11): *The success login message.*

The figure (4.12): Shows the main screen system, with a welcome message to the authenticated user who has access to the system content.

Figure (9.3): *The main system screen.*

5. CONCLUSION

This application is designed to increase the level of protection, by taking advantage of the security of Android, activated by putting it in the framework of a specific practical model that helps, the user to authenticate themselves in a simple and unique way, through mobile phone.

The paper concluded that the design of an Android-based mobile phone application, taking advantage of the security features of the Android system, enables the paper to achieve highly satisfactory results, with multiple additional protection levels, which in turn improves access to content, and only authenticating the authenticated users.

It remains, there is no security at whole, but it is an attempt to increase reliability, make systems content a distance from threats and blatant abuse.

A. Future Works

There are several things that app developers can take advantage of and exploit to add new app leverage, for example:

- Develop the idea of mobile authentication to access cloud computing content.
- Modifying the idea of authentication for using the mobile phone in Internet of Thing (IoT) accessing.
- Create special purpose devices that operate in the same way as to control the actual incomes of employees in organizations.
- Enhance the idea to control attendance, absence of lectures and conferences etc.

B. Recommendation

Recommendations can be summarized as follows:

- Finding more convincing and unique solutions to the padding.
- Improve the rate of authentication transaction, by reducing the process time to the minimum as well as possible.
- Find appropriate ways to take advantage of the features of the Android system, (GPS, encryption, etc.), within the application to raise its efficiency to the maximum.

REFERENCES

[1] Anderson, J.M., 2003. Why we need a new definition of information security. *Computers & Security*, *22*(4), pp.308-313.

[2] Whitman, M.E. and Mattord, H.J., 2011. *Principles of information security*. Cengage Learning.

[3] Crowley, E., 2003, October. Information system security curricula development. In *Proceedings of the 4th conference on Information technology curriculum* (pp. 249-255). ACM.

[4] Friesen, J.J., 2010. *Getting Started with Java*. Learn Java for Android Development, pp.1-41.

[5] *Android System Development* 2004-2017, accessed 03 March 2017, <http://www. free-electrons.com/doc/training/ android>.

[6] Al-Sinani, H.S. and Mitchell, C.J., 2011, June. Enhancing CardSpace Authentication Using a Mobile Device. In *DBSec* (pp. 201-216).

[7] Mustafa, A.F. and Ja'afer, A.S., 2011. An enhancement of authentication protocol and key agreement (AKA) for 3G mobile networks. *International Journal of Security (IJS)*, 5(1), pp.35-51.

[8] Vishal, G., Ravishanker and and Ashish, Kr.L, 2016. Mobile Based Secure Authentication Using TLS and Offline OTP. *International Journal of Computer Technology and (IJCTA)*, 9(11), pp. 5253-5262.

[9] Liao, I.E., Lee, C.C. and Hwang, M.S., 2005, August. Security enhancement for a dynamic ID-based remote user authentication scheme. In *Next Generation Web Services Practices, 2005. NWeSP 2005. International Conference on* (pp. 4-pp). IEEE.

[10] Das, M.L., Saxena, A. and Gulati, V.P., 2004. A dynamic ID-based remote user authentication scheme**.** *IEEE Transactions on Consumer Electronics*, *50*(2), pp.629-631.

[11] Gong, J. and Tarasewich, P., 2004, November. Guidelines for handheld mobile device interface design. In *Proceedings of DSI 2004 Annual Meeting* (pp. 3751-3756).

[12] Stüber, G.L., 2001. *Principles of mobile communication* (Vol. 2). Boston: Kluwer Academic.