Modular Biometric Authentication Service System (MBASSy)

Heiko Witte, Claudia Nickel
Hochschule Darmstadt*
heiko.witte@cased.de, c.nickel@fbi.h-da.de

Abstract: In this paper we present the design of a modular authentication system, which enables users to select an authentication procedure by preference.
A survey, carried out by N.L. Clarke and S.M. Furnell [CF05], proved the classical PIN-authentication to be inconvenient for many users. Passwords and PINs are either secure or easy to remember. Since humans tend to forget complex permutations of characters and numbers, the chosen secrets are often insecure.
The purpose of the system is to provide an infrastructure which allows the implementation and usage of alternative authentication procedures. This approach could lead to an increased acceptance of authentication on smartphones and thus an increased security of the devices. A prototype based on the concepts presented in this paper was implemented for the android operating system and a gait recognition module is being actively developed. Further modules like face recognition, voice recognition or graphical authentication schemes can be integrated which depicts the flexibility of the system.

1 Introduction

With the growing amount of smartphones in use, an increased demand in information security arises. The classical approach of user authentication relies on knowledge-based methods, with the PIN being the common implementation among these. Many users tend to forget passwords and PINs, which leads to increased efforts that the majority of users avoid. This decreases the security of the devices.
In this paper, we present a prototype of a modular authentication system for the android operating system. Authentication algorithms are outsourced into distinct application packages, which either require interaction with the user or run in the background. The system was developed to facilitate the deployment of biometric authentication algorithms. Biometrics provide alternative ways to authenticate a user, which may reduce user effort and thus increase the acceptance of authentication on mobile devices.
The modular design of the system allows the user to select the kind of authentication he prefers, thus encouraging the development of alternatives to the classical PIN authentication. The ability to activate multiple modules enables further extensions of the system.

*This work was supported by CASED (www.cased.de)

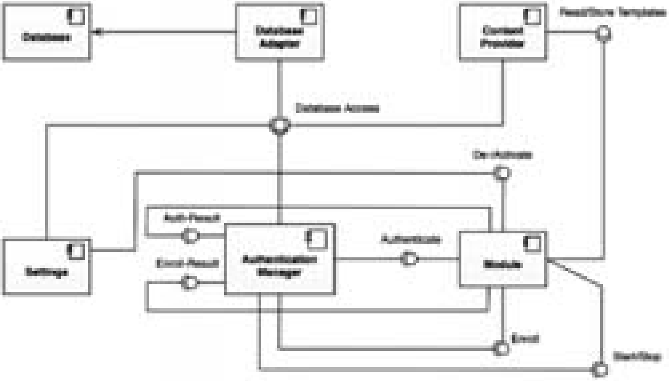

Figure 1: System overview

2 Components of MBASSy

The Modular Biometric Authentication Service System (MBASSy) was developed for the android operating system. Information on the android application fundamentals may be obtained from the extensive online documentation [Goo10]. All android specific terms in this paper are in *italics*. The system design follows most of the guidelines described in a publication by Saltzer and Schroeder [SS75]. The system consists of the following four main components: Database, Modules, User Interface, Background Service.

2.1 Modules

Modules contain a comparison function and the methods to obtain and process the user input. Two distinct types of modules are supported by MBASSy:

Background Modules run in a background process and do not require interaction with the user during the authentication process.

Foreground Modules require interaction with the user in all cases.

An interface shall be implemented to provide accessor methods for MBASSy. The interface is necessary to perform the following actions: ACTIVATE_MODULE, DEACTIVATE_MODULE, AUTHENTICATE, ENROLL, START (Background Modules), STOP (Background Modules).
When a module receives the activation action, a token is passed in the received *Intent*. This token must be used in every access request to the database. It shall be saved in the application package for later reference. If the token is lost, the module will lose access to its data.

When the enrollment action is invoked in the module interface, an authentication token is passed to the module in a data structure. The authentication token must be used in combination with the module token to gain write access in the *ContentProvider*.

2.2 Database

A database adapter class provides accessor methods for internal usage within the application package. A *ContentProvider* exposes a restricted version of those accessor methods to the modules. Read access is possible at any time, while write access is only granted in case of a pending enrollment request. This restriction was imposed on the modules to stay in control of the database size. In both cases, modules must supply their token and, in case of requesting write access, the authentication token which was supplied by the authentication service.

Read access is always available through a standard Java *InputStream* by constructing a URI of the following form:

```
content://com.cased.biometrics.provider.mbassy/ \
biometric_data/MODULE_TOKEN/TEMPLATE_ID
```

The template identifiers are part of the result set of a *ContentResolver* query. When requesting write access, the above URI is used without the appended template identifier. The required authentication token, as well as the user identifier, module token and user data are passed to the insert method of the *ContentResolver* in a *ContentValues* object. The user data shall be provided as a byte array. The *ContentProvider* takes care of writing the data to a file and storing the appropriate URI in the database.

All data is stored in the form provided by the module. The system does not encrypt or otherwise ensure the security or integrity of the data. Thus, modules shall provide ways to protect the data. The applicable protection algorithm may vary depending on the type of authentication algorithm. In the case of biometrics, biometric template protection is an appropriate way to ensure privacy [BBGK08].

2.3 User Interface

The user interface consists of the system preferences, account settings, module settings and system information overview. The module settings are an integral part of the system and will be discussed in detail.

When opening the module settings, a broadcast is sent to the operating system. The *Action* of the *Broadcast Intent* is **GET_MODULE_INFO**.

The *BroadcastReceiver* of a module is activated by the system and a response is generated by setting up an *Explicit Intent* which is sent back to the module settings. The following information of a module is passed in this *Intent*: Module name, Module type, Full package name, Full package path to the class implementing the required interface, Module token (if available). The action of the *Intent* is **SET_MODULE_INFO**. If a module token is found in the database, it is displayed in the ”Active Modules” section, otherwise in the ”Available Modules” section. Modules can be activated, deactivated and prioritized in the module settings. The information on activated modules and their priorities is stored in the database. The priority is used by the database adapter to provide an ordered list of modules to the authentication service, according to the users preference.

2.4 Background Service

The background service constitutes the core of MBASSy. The service is responsible for managing authentication and enrollment requests. A broadcast receiver is used to take action upon receiving system broadcasts for the events ”screen off”, ”screen on” and ”battery low”.

When the service receives a ”screen off” event, a timer is started, which locks the device after a user defined period of time. When MBASSy is in a locked state and receives the system event ”screen on”, the background service fetches a list of active modules from the database and sequentially sends authentication requests to them.

MBASSy supports two authentication modes. The ”Single Module Mode”, requires a positive authentication result of a single module to successfully complete the authentication process. The ”All Modules Mode” requires positive authentication results of all active modules in the system. In this early stage of development, modules return match decisions rather than comparison scores. The systems capabilities will be extended in the future by providing an authentication mode which accepts comparison scores as input and makes authentication decisions based on score level integration [BCP$^+$04].

The background service stores log messages for every authentication or enrollment process in the database. This data may be used to learn about the system and user behaviour or the performance of modules. The background service performs the following tasks:

Authentication Requests An authentication request is performed by setting the respective *Action* of an *Explicit Intent*. The background service generates a unique authentication token, which shall be returned by the module when completing the request. The result is discarded if the returned authentication token does not match the one issued by the service. The data of an authentication result is also stored in an *Explicit Intent*, which contains at least the result as a boolean value, the authentication token and, in the case of a match, an identifier of the matched reference template as *Extra* values. Future versions of MBASSy will accept a comparison score as *Extra* value. The authentication process is depicted as a sequence diagram in figure 2.

Enrollment Requests Enrollment requests are issued automatically by the system when a new user account is created or when the user manually enrolls for a specific module in the account settings. The communication between MBASSy and the module is similar to the authentication process.

MBASSy stores the authentication token using the *SharedPreferences* of android, which are also accessible by the *ContentProvider*. A module shall pass the authenti-

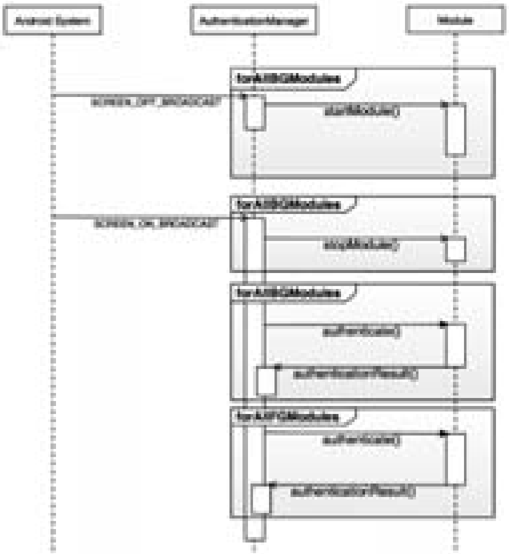

Figure 2: The authentication process

cation token as described in section 2.2. Write permission is denied without a valid combination of module token and authentication token. The enrollment process is depicted as a sequence diagram in figure 3.

Lock Screen Android provides no function for third party developers to securely lock a device. Due to this lack of functionality, a workaround was implemented. The workaround disables keys on the smartphone and replaces the home screen of android with a lock screen. As soon as Google releases a SDK which provides an official function to lock a device, it will be adopted in MBASSy.

3 Conclusion

Due to the flexibility of MBASSy, a combination of different authentication types is possible like e.g. knowledge-based authentication with biometric authentication. This could be a PassShape [WL08] module in combination with a gait recognition module. Fusion of multiple biometric modalities can also be accomplished by activating the respective authentication mode. Other use cases include the development of modules that collect biometric samples to create databases for testing purposes.

Authentication systems like the Local Authentication Subsystem (LASS) of Windows Mobile [Mic10] do not provide the functionality of MBASSy. Even though LASS supports the deployment of alternative authentication algorithms in a DLL, it does not support the activation of multiple modules.

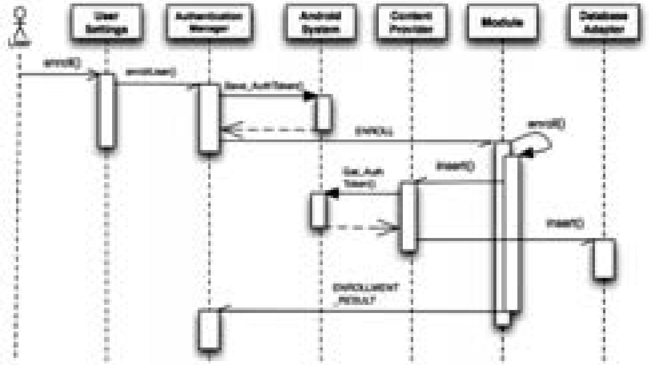

Figure 3: The enrollment process

References

[BBGK08] Jeroen Breebaart, Christoph Busch, Justine Grave, and Els Kindt. A Reference Architecture for Biometric Template Protection based on Pseudo Identities. *BIOSIG 2008 - Proceedings of the Special Interest Group on Biometrics and Electronic Signatures*, pages 25–38, 2008.

[BCP$^+$04] Ruud M. Bolle, Jonathan H. Connell, Sharath Pankanti, Nalini K. Ratha, and Andrew W. Senior. *Guide to Biometrics*. Springer, 2004.

[CF05] N.L. Clarke and S.M. Furnell. Authentication of users on mobile telephones - A survey of attitudes and practices. *Computers Security*, pages 519–527, 2005.

[Goo10] Google. Android Application Fundamentals. `http://developer.android.com/guide/topics/fundamentals.html`, March 2010. 14.03.2010.

[Mic10] Microsoft. Local Authentication Subsystem (LASS). `http://msdn.microsoft.com/en-us/library/aa923670.aspx`, April 2010. 11.05.2010.

[SS75] J.H. Saltzer and M.D. Schroeder. The Protection of Information in Computer Systems. *Proceedings of the IEEE*, 63(9):pages 1278–1308, March 1975.

[WL08] Roman Weiss and Alexander De Luca. PassShapes: utilizing stroke based authentication to increase password memorability. *ACM International Conference Proceeding Series. Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, 358:383–392, 2008.