DEPLOYING PERSONALIZED OBLIVIOUS MOBILE IDENTITY

Georgia Sourla[1,2] and Evangelos Sakkopoulos$^{2[0000-0002-6852-384X]}$

[1] University of Patras, Computer Engineering and Informatics Department, Greece
[2] Scytales, Sweden gsourla@scytales.com
[3] University of Piraeus, Department of Informatics, Piraeus, Greece
sakkopul@unipi.gr

Abstract. Mobile Id approach provides a contactless and secure identification process. Standardization towards a common scheme is important to ensure wide-spread adoption through cross-country and cross-region interoperability. Therefore mobile identification has seen enormous attention both by platform vendors and researchers. In this work we discuss a novel mobile identification approach that is based on oblivious identity to ensure improved privacy protection and identification data at rest protection. The mobile identification approach addresses the development of the interfaces to connect users' identities to their real-world physical identities by integrating with officially issued eIDs through bridging with eIDAS, STORK, or electronic driving licenses. In particular we present the initial validation for mobile identification of distributed virtual iDP cryptographic infrastructure for mobile ID use cases. The key aim is to show the particular steps needed in order to perform thorough testing in new cryptographic approaches that aim to be applied in real life application through presented first pilots. Evaluating these prototypes and also evaluating them with other stakeholders in order to identify a wider deployment for business development and exploitation. Initial testing and validation results of the proposed mobile identification approach are encouraging and promising showing strong privacy preservation outcomes.

Keywords: mobile ID · digital identity · oblivious design.

1 Introduction

Identification through mobile devices is possible to support secure and safe id verification both in person or in remote based approaches using appropriate architectures. Lots of focus on face to face verification identity has been given in the recent years. Towards the seamless connectivity and communication of mobile id approaches, standardization plays an outstanding role to allow interoperability while ensuring key security measures and risk mitigation approaches. Identity verification is used for several occasions everyday. Proving identity fully or partially is needed to access age controlled products (i.e. tobacco), to pick-up registered mail and parcels, to perform physical access control in facilities and buildings and it can even be applied in most recent flow control events as to allow traffic during COVID-19 restriction measures world-wide [1].

Therefore bringing ID into the mobile has seen enormous attention both by industry and academia using a number of different approaches [10]. In this work, we discuss a novel mobile identification approach that is based on oblivious identity [9] to ensure improved privacy protection and identification data at rest protection [7]. The mobile identification in this work addresses the development of the interfaces to connect users' identities to real-world physical identities by integrating with officially issued eIDs through bridging with eIDAS, STORK, or electronic driving licenses.

This work studies the real working pilots design and approach. The aim is to analyze the possible deployment approaches and the initial working concepts of the mobile solution of corresponding to a mobile identification use case. We include the initial feedback to confirm deployment is aligned to the requirements defined from the real-world situations.

In the online approach, the holder of the mobile Driver Licence (mDL) must be online during the whole process of both registration and verification. On the other hand, the mDL verifier must be online only during the setup phase, in order to get the public key from the virtual Idenity Provider (vIdP) [6]. Then the verification process shall be performed in offline mode, since no other information is needed from the vIdP. From the mDL holder's perspective, we suppose he already possesses a digitally signed mDL on his mobile device (i.e. smartphone, tablet, smart POS), which's authenticity is certified by the corresponding Issuing Authority.

In the offline scenario, the mDL holder is online during the registration process but goes offline during the verification process. The same holds for the mDL verifier, since his device must be online during the setup phase, but the verification process is made in offline mode. From the mDL holder's perspective, we suppose he already possesses a certified mDL on his mobile device. [6].

2 Related Work

Privacy Attribute-Based Credentials (P-ABCs) Figure 1 [9] presents a possible algorithmic approach to achieve pseudo-anonymity and minimum disclosure of data. P-ABCs are based on the similar "offline" architecture as X.509 digital certificates. A key difference between X509 certificates and P-ABC credentials is that the latter provide a tool to derive specialized one-time tokens that only reveal the minimum needed number of pieces of information [9].

An advanced design for P-ABCs are distributed P-ABCs (dP-ABC). The dP-ABC Credential Management module on the IdP needs to manage the cryptographic material for the scheme. Moreover, it provides the credentials for each user. The analysis of the cryptographic techniques that are being developed for distributed P-ABCs in ObLivious Identity and Management for Private and User-friendly Services are presented further in [3]. ObLivious Identity and Management for Private and User-friendly Services [9] are privacy-preserving identity

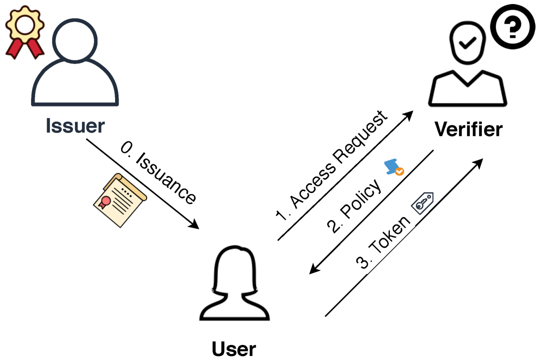


Fig. 1. p-ABC Architectural Approach

provision services that use distributed cryptography to distribute the functionality and cryptographic computations of the IdP across multiple authorities, so that no single authority can impersonate or track its users.

Although proposals such as Identity Mixer [8], Distributed Password [7] or European projects such as ARIES [5] represent promising and efficient approaches for the deployment of P-ABC systems, their adoption remains to be seen [4].

3 Online Distributed Token Approach

The workflow from the mDL holder's and mDL verifier's perspective can be described using the following step-by-step approach in the online case.

VERIFIER SETUP:

For setting up the mDL verifier application, the verifier needs to start it and press the setup button, in order to obtain the public key from the vIdP server. Once the key is successfully received, the mDL verifier application is ready to perform verifications.

VERIFICATION:

A verification procedure may start when the mDL verifier initiates a connection with the mDL holder's device, in order to send the policy that the holder needs to fulfill. When the policy is received by the mDL holder application, the mDL holder will see which attributes shall be revealed to the mDL verifier. If the mDL holder chooses to accept the policy, a request is performed from the mDL holder application to the vIdP, in order to provide the mDL holder with a token for the specified policy. After receiving the token, the mDL holder application forwards the token to the mDL verifier, so that he can proceed with the verification process. Once the token is sent from the mDL holder application, the mDL verifier can verify it against the public key obtained during setup. In the end, a message about the result of the verification appears on the mDL verifier application screen.

The main advantage of this approach is that during every verification, the most updated identification attributes are used in order to generate a token, so the mDL verifier can be sure that he verifies the mDL holder against updated personal information. On the other hand, this approach requires that the mDL holder has an online connection established throughout the whole verification procedure.

4 Offline scenario dP-ABC credentials

The offline case is quite different, since only the setup phase requires an online connection. As a result, the mDL verifier application is setup with the same steps, described in the previous section. The verification procedure though is now performed following the process described below.

VERIFICATION:

In an offline verification, the mDL verifier initiates a connection with the mDL holder's device, in order to send the policy that the holder needs to fulfill. When the policy is received by the mDL holder application, the mDL holder will see which attributes shall be revealed to the mDL verifier. Up to this point, the procedure follows the same steps as in the online scenario. This time though, if the mDL holder chooses to accept the policy, it is the mDL holder application and not the vIdP that shall generate a token based on the credentials acquired from enrolment procedure and the policy received from the verifier. After generating the token, the mDL holder application forwards it to the mDL verifier, in order to proceed with the verification process. The mDL verifier receives the token from the mDL holder application and verifies it against the public key obtained during setup. As in the online scenario, the verifier needs no further communication with the vIdP, after acquiring the public key, so the whole verification procedure is performed in offline mode. In the end, a message about the result of the verification appears on the mDL verifier application screen.

With this approach, there is no need for the mDL holder to be connected during verification, thus he can be verified and make use of the requested service at any time needed. On the other hand, the mDL verifier can perform verification based on the personal information that is currently stored in the mDL holder's device. We expect though, that the mDL holder updates his info in a regular base, in order to avoid expiration.

5 Architecture and Integration

The following sections describe how the integration was performed between each component and the current state overall.

5.1 Interfaces

The mDL applications for both holder and verifier need to communicate with the vIdP. From the holder's side, there is the need to create an account to the vIdP which includes the holder's personal information, that would be available for authentication, depending on the verification requirements. There is also the need for token creation, that would be generated either directly from the vIdP in the online scenario or from the mDL holder in the offline scenario. The latter is possible by utilizing the dP-ABC Client component, which can store the mDL holder's credentials, obtained from the vIdP during enrolment, and use them anytime later in order to create tokens for the specified policies in offline mode. The interfaces used for the aforementioned procedures can be viewed in Figure 2 and Figure 3.

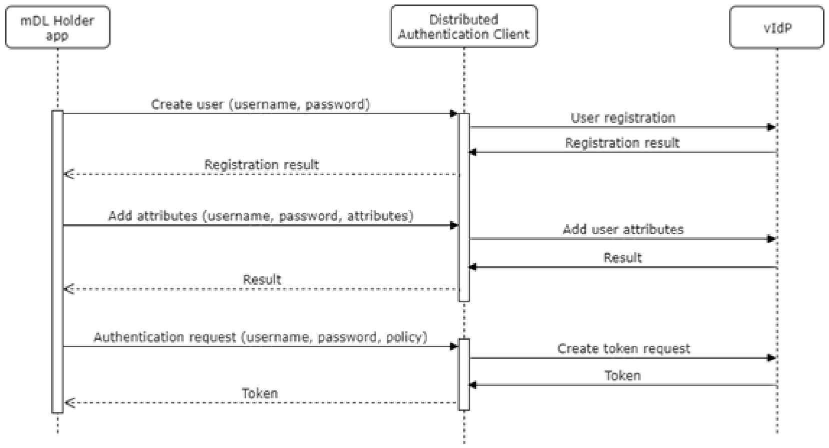

mDL Holder app
Distributed Authentication Client
vIdP
Create user (username, password)
User registration
Registration result
Registration result
Add attributes (username, password, attributes)
Add user attributes
Result
Result
Authentication request (username, password, policy)
Create token request
Token
Token

Fig. 2. Interfaces for mDL Holder (online scenario)

From the verifier's side, either for the online or for the offline scenario, the only connection needed to be established with the vIdP, is made on the setup phase for acquiring the public key that shall be used during verification.

5.2 Components

From the previous section, we can easily understand that there are quite a number of modules being used for the mDL use case. The two basic ones for this testing scenario are the mDL holder application and the mDL verifier application. These two applications allow us to demonstrate the mDL use case by pretending to have a real mDL holder that is being verified against a certain policy (i.e. proof that the holder's age is above a certain limit) by an mDL verifier. Currently two different systems are supporting the online and the offline scenario.

In the online scenario, the Distributed Authentication Client module is utilized in order to perform registration and authorization. This module establishes a connection with the vIdP server in order to create a new user account, store the

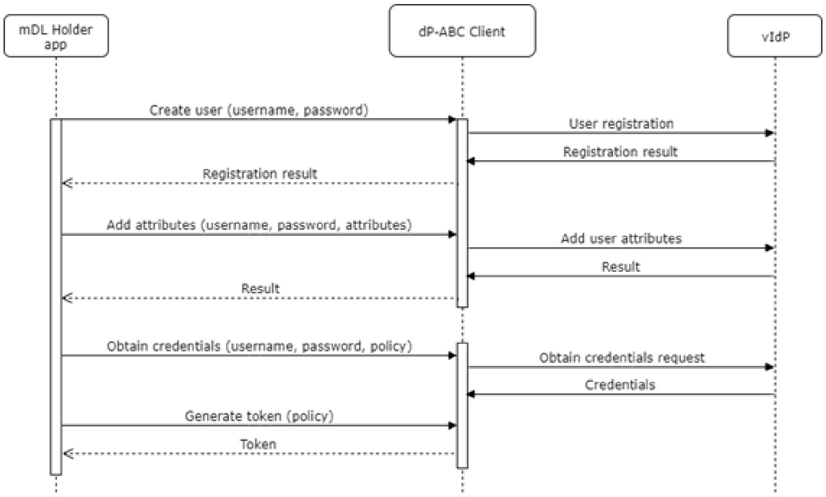


Fig. 3. Interfaces for mDL Holder (offline scenario)

mDL holder's personal information and provide to the holder an authorization token compliant to a specific policy. In the offline scenario, this part is played by the dP-ABC Client component. The main difference between these two modules is that the dP-ABC Client component needs to store the mDL holder's P-ABC credentials, in order to be able to generate tokens in offline mode at a later point in time. On the contrary, the Distributed Authentication Client module does not need to store information about the mDL holder and directly forwards the generated token from the vIdP to the mDL holder application.

6 Design and development of components

Currently, for the mDL use case we have been focusing on the offline scenario for the first pilot deployment. We are expecting though that dealing only with the offline scenario will pay back when we turn to the online one, not only because of the feedback that would be received from the first demo testing, but also from all the issues that already have been overcome through this initial offline implementation of the mDL use case.

6.1 The mDL Issuing Authority

The mDL Issuing authority role is performed by an instance of the vIdP.

As an initial vIdP pilot deployment process for the offline scenario, three instances of the Proactively Secure Distributed Single Sign-On (PESTO) IdP were deployed as an embedded servlet container. These three instances would provide distribution during all mDL holder's actions, such as enrollment, adding identification attributes and issuing credentials. Additionally, all instances are aware of the existence of the rest IdP instances and each one of them is configured with a JSON file that is used during each PESTO IdP startup phase.

For the offline scenario pilot, the mDL holder application connects to the three IdPs via dP-ABC client interface, while the mDL verifier application utilizes the dP-ABC verifier interface in order to connect to the vIdP. These two interfaces provide all the needed functionalities from a single point, and as a result both mDL applications are able to perform client registration, identity attachment, credential issuance and token verification, accordingly.

6.2 The mDL Apps

For the offline mDL scenario, two applications have been implemented, one for the mDL holder and one for the mDL verifier. They are implemented in such a way that the mDL holder application initiates the mDL verifier application while sending the authentication token.

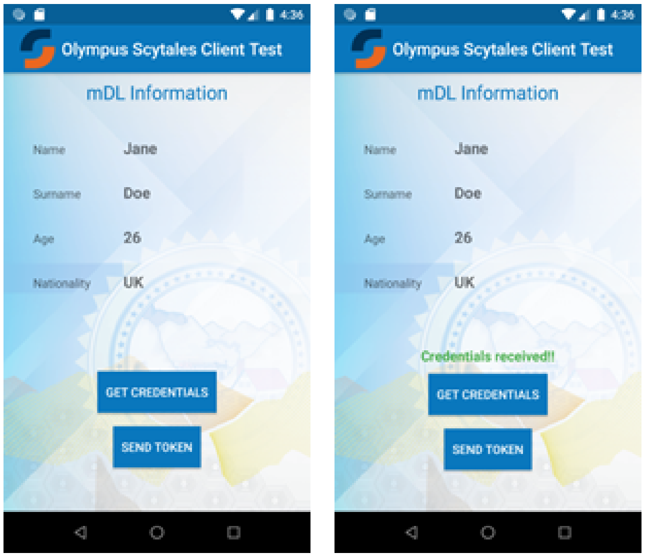
With the mDL holder application, the holder is able to create his own account at the vIdP, by registering with a username and password of his choice. The implemented application gathers the typed credentials and the mDL holder's personal information from the mDL he possesses and sends all needed data to the vIdP, in order to create an account and store the holder's personal info. The result of this procedure is shown to the mDL holder with an appropriate message.

7 Evaluation

The following sections describe how the demonstrations that took place for the offline scenario should be replicated, as well as the testing plan for the mDL use case. In the experimental prototype section functionality testing is presented to show that the basic necessary mDL use case functions are covered focusing on mDL initialization (also called sign-up) and the verification process for valid and invalid cases. Next, unit testing and integration testing is described, where the mDL has been interconnected and adapted to use distributed and oblivious approach. We have verified that the all modules are communicating successfully and in a valid manner.

7.1 Functional Verification of the Experimental Prototype

For demonstrating the mDL use case, two Android mobile applications were developed, one for the mDL holder and one for the mDL verifier. The mDL holder application needs to be online during the enrollment phase. We assume that the holder already possesses a digitally signed mDL on his device. After a successful registration, the mDL holder application presents the holder's personal information on the screen. The holder is able to request his P-ABC credentials from the vIdP and go offline (Figure 4). Now he can generate at any time needed the authentication token which fulfils the policy that the mDL holder application supports. This policy is checking the name and the age of the mDL holder for a standardized age verification process.



mDL Information

Name Jane

Surname Doe

Age 26

Nationality UK

mDL Information

Name Jane

Surname Doe

Age 26

Nationality UK

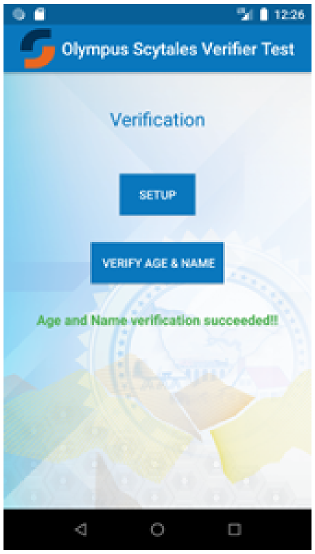
Credentials received!!

Fig. 4. mDL Holder app - Credentials Request and Result

The mDL verifier application needs to be online only during the setup phase. When the verifier starts the verifier application, he just needs to press the "Setup" button, in order to acquire the public key from the vIdP and prepare his application for verification procedure. After that, he can go offline and wait for a token to be received.

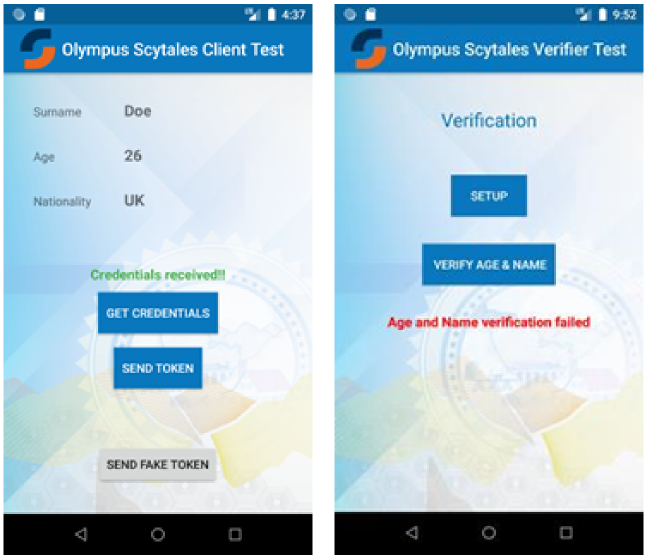
In order for the holder to be verified, he needs to press the "Send Token" button in the mDL holder application, in order to generate a token for the supported policy and forward it to the verifier. After this action, the mDL verifier application appears and the verifier may now press the "Verify" button. The verification procedure shall start, in order to check the name and the age of the mDL holder and the result will appear on the screen with a proper message (Figure 5).

Within demo scope, there is one more button added in the mDL holder application that simulates the case of a fraud holder. What this button does is to alternate the signature of the P-ABC credentials received from the vIdP before generating the authentication token. If this button is pressed during verification process, the mDL verifier application shall show a message of failed verification (Figure 6).



Age and Name verification succeeded!!

Fig. 5. mDL Verifier app - Successful age and name verification



Surname Doe

Age 26

Nationality UK

Credentials received!!

Age and Name verification failed

Fig. 6. mDL Holder and Verifier app - Failed age and name authentication (b)

7.2 Unit and Integration testing plans

Currently the testing plan for the mDL use case involves unit and integration testing. As a first step, unit tests were created in order to test small parts of implementation for the two mDL applications under development. At the moment, we have reached a stage where servers are hosting the vIdP integrated system and the mDL applications are connected to them in order to perform the registration and verification procedures.

Unit tests were developed in two phases. At first, simple parts of the implementation were developed along with the user interface, in order to have a basic skeleton for our applications. During this phase, no communication was supported with external systems. On a next level, a first stable version of the new cryptographic system proposed in OLYMPUS was available, with support of dP-ABC credentials. The whole process of registration and verification was tested, with the cryptographic modules being set up in a local server. The same process was also followed when the whole Olympus integrated vIdP system was hosted in one partner's server for the demo tests.

In all the testing steps mentioned above, both successful and failed scenarios were tested, so that the integrity of not only the two developed mobile applications, but also of the whole Olympus system would be tested. The results of unit and integration tests have been positive and promising.

8 Conclusions and Future Work

As we reached the middle of the timeline of the project, there has been a lot of work being done for the development of not only the new cryptographic systems proposed in this project, but also the demo pilots that shall be used for the testing of the aforementioned systems. Yet, there are further steps to be done for having a fully functional integration of the mDL pilot with the rest of the system, whose development is still ongoing.

A further addition would be to securely store data in the devices of the mDL holder and the mDL verifier. With this extension, the two mobile applications shall have all required data, which were already acquired, available for later usage.

Last but not least, the online version of distributed token shall also be supported in the mDL pilot at a later time. With this extension, the mDL holder will be able to get online the token that complies with the policy specified by the mDL verifier he encountered and pass it straight ahead to the verifier's device in order to proceed with the verification.

Acknowledgment

TBD This approach has also received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 786725 (OLYMPUS project).

References

1. Covid-19 restrictions. FRONTEX: European Border and CoastGuard Agency (2020), https://frontex.europa.eu/media-centre/news-release/covid-19-restrictions-4IdY3J

2. Agrawal, S., Miao, P., Mohassel, P., Mukherjee, P.: PASTA: password-based threshold authentication. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018. pp. 2042–2059 (2018), https://doi.org/10.1145/3243734.3243839
3. Baum, C., Frederiksen, T.K., Hesse, J., Lehmann, A., Yanai, A.: PESTO: proactively secure distributed single sign-on, or how to trust a hacked server. IACR Cryptology ePrint Archive p. 1470 (2019), https://eprint.iacr.org/2019/1470
4. Bernabé, J.B., Cánovas, J.L., Ramos, J.L.H., Moreno, R.T., Skarmeta, A.F.: Privacy-preserving solutions for blockchain: Review and challenges. IEEE Access **7**, 164908–164940 (2019), https://doi.org/10.1109/ACCESS.2019.2950872
5. Bernabé, J.B., David, M., Moreno, R.T., Cordero, J.P., Bahloul, S., Skarmeta, A.F.: ARIES: evaluation of a reliable and privacy-preserving european identity management framework. Future Gener. Comput. Syst. **102**, 409–425 (2020), https://doi.org/10.1016/j.future.2019.08.017
6. Camenisch, J., Drijvers, M., Lehmann, A., Neven, G., Towa, P.: Short threshold dynamic group signatures. IACR Cryptology ePrint Archive **2020**, 16 (2020), https://eprint.iacr.org/2020/016
7. Camenisch, J., Lehmann, A., Neven, G.: Optimal distributed password verification. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015. pp. 182–194 (2015), https://doi.org/10.1145/2810103.2813722
8. Camenisch, J., Mödersheim, S., Sommer, D.: A formal model of identity mixer. In: Kowalewski, S., Roveri, M. (eds.) Formal Methods for Industrial Critical Systems - 15th International Workshop, FMICS 2010, Antwerp, Belgium, September 20-21, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6371, pp. 198–214. Springer (2010), https://doi.org/10.1007/978-3-642-15898-8_13
9. Moreno, R.T., Bernabé, J.B., Rodríguez, J.G., Frederiksen, T.K., Stausholm, M., Martínez, N., Sakkopoulos, E., Ponte, N., Skarmeta, A.F.: The OLYMPUS architecture - oblivious identity management for private user-friendly services. Sensors **20**(3), 945 (2020), https://doi.org/10.3390/s20030945
10. Sakkopoulos, E., Ioannou, Z., Viennas, E.: Mobile personal information exchange over BLE. In: 9th International Conference on Information, Intelligence, Systems and Applications, IISA 2018, Zakynthos, Greece, July 23-25, 2018. pp. 1–8. IEEE Computer Society (2018), https://doi.org/10.1109/IISA.2018.8633599
11. Sanchez, J.L.C., Bernabé, J.B., Skarmeta, A.F.: Integration of anonymous credential systems in iot constrained environments. IEEE Access **6**, 4767–4778 (2018), https://doi.org/10.1109/ACCESS.2017.2788464