Killing the Password and Preserving Privacy with Device-Centric and Attribute-based Authentication

Kostantinos Papadamou*, Savvas Zannettou*, Bogdan-Cosmin Chifor†, Sorin Teican†, George Gugulea†
Annamaria Recupero◇, Alberto Caponi‡, Claudio Pisa‡, Giuseppe Bianchi‡, Steven Gevers∓
Christos Xenakis±, Michael Sirivianos*

*Cyprus University of Technology, †certSIGN, ◇Sapienza University of Rome, ‡University of Rome Tor Vergata,
∓Verizon Enterprise Solutions, ±University of Piraeus
{ck.papadamou, sa.zannettou}@edu.cut.ac.cy, {bogdan.chifor, sorin.teican, george.gugulea}@certsign.ro,
annamaria.recupero@gmail.com, {alberto.caponi, claudio.pisa, giuseppe.bianchi}@uniroma2.it,
steven.gevers@be.verizon.com, xenakis@unipi.gr, michael.sirivianos@cut.ac.cy

Abstract

Current authentication methods on the Web have serious weaknesses. First, services heavily rely on the traditional password paradigm, which diminishes the end-users' security and usability. Second, the lack of attribute-based authentication does not allow anonymity-preserving access to services. Third, users have multiple online accounts that often reflect distinct identity aspects. This makes proving combinations of identity attributes hard on the users.

In this paper, we address these weaknesses by proposing a privacy-preserving architecture for device-centric and attribute-based authentication based on: 1) the seamless integration between usable/strong device-centric authentication methods and federated login solutions; 2) the separation of the concerns for Authorization, Authentication, Behavioral Authentication and Identification to facilitate incremental deployability, wide adoption and compliance with NIST assurance levels; and 3) a novel centralized component that allows end-users to perform identity profile and consent management, to prove combinations of fragmented identity aspects, and to perform account recovery in case of device loss. To the best of our knowledge, this is the first effort towards fusing the aforementioned techniques under an integrated architecture. This architecture effectively deems the password paradigm obsolete with minimal modification on the service provider's software stack.

1 Introduction

Authentication in the web relies on the password paradigm, which was developed during the 60s for accessing monolithic mainframe computers. We admit that a 128-bit very complex and long (∼20 characters) password used for a specific service is highly secure when it is only stored in the brain of the user and it is computationally hard to guess. However, as the needs and number of web services increases, the password paradigm entails an inextricable tension between security and usability as users become burdened with memorizing and managing multiple passwords. At the same time, passwords can be shoulder-surfed, key-logged, replayed, eavesdropped, brute-forced and phished. In addition, password databases can be leaked and even if the service follows security good practices (i.e., hashing and salting the passwords), the attacker can guess the password by performing a dictionary-based brute-force attack. Over the years, the scientific community repeatedly pinpointed the flaws of the password paradigm [7, 28, 4, 17].

Fig. 1 depicts the three main caveats of the currently prevalent web authentication paradigm. First, the password overload problem where users need to remember one secure password for each service. As a consequence, users resort in re-using the same password for each service they maintain [20]. Second, there is lack of support for Attribute Based Access Control (ABAC), which facilitates account-less authentication through identity attributes (i.e., age or location). Last, a user's identity is fragmented across multiple services. This renders the task of proving account joint-ownership of services hard for end-users.

Recent efforts aim at mitigating the aforementioned problems by proposing dedicated solutions. Specifically: 1) federated authentication solutions (i.e., OpenID Connect[1]) alleviate the password overload problem by enabling a Service Provider (SP) to delegate the authentication of end-users to a trusted entity called Identity Provider (IdP); 2) strong and usable password-less authentication mechanisms, such as FIDO UAF[2]; and 3) cryptographic credential stacks that facilitate Privacy-preserving Attribute-based Access Control (PABAC) such as Idemix [8] and U-Prove[3]. Despite the fact that the aforementioned solutions mitigate the problems to some extent, they suffer from deployability issues as SPs are required to deploy multiple specialized components within their infrastructure.

Other studies [30, 24, 2] propose the use of password managers, which enable the user to use distinct strong passwords for each online service they use, while the burden of maintaining and remembering the password is offloaded to the password

[1] http://openid.net/connect
[2] https://fidoalliance.org/specifications/overview
[3] https://www.microsoft.com/en-us/research/publication/u-prove-cryptographic-specification-v1-1-revision-3/

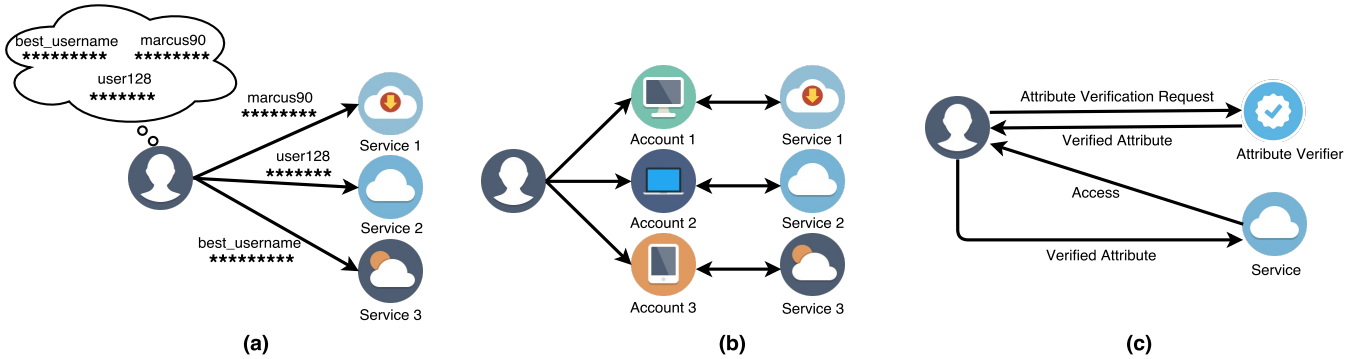

Figure 1: Caveats of the prevalent web authentication paradigm: 1) password overload; 2) identity fragmentation; and 3) lack of support for Attribute-based Access Control.

manager. However, unlike device-centric authentication with FIDO public-key cryptography, password managers still rely on secret tokens. Therefore, password managers are susceptible to online guessing, replay, session hijacking, eavesdropping and breach attacks.

In this work, we propose a privacy-preserving federated architecture for device-centric authentication (DCA) that aims to anchor all users' access control needs to devices (i.e., smartphones) that they habitually carry along. "Something that endusers almost always have with them", allows users to not have to always "know something for all those accounts they maintain", thus solving the password overload problem.

Moreover, DCA requires special authenticators that most SPs do not have. Following recent industry trends, we propose the integration of the design elements proposed by the FIDO Alliance for strong authentication mechanisms, and from the OpenID Foundation for federated authentication. This integration enables a federated authentication solution where endusers are able to authenticate using biometrics. The main advantage of this approach is that the core authentication functionality resides on a trusted entity (IdP), and services (SPs) are able to incrementally adopt this approach with minimal modifications to their infrastructure.

DCA and federation enables the enclosure of strong cryptographic protocols transparent to the user within the device, thus seamlessly supporting anonymity-preserving attribute-based authentication. Additionally, the various sensors embedded in mobile devices facilitate behavioral authentication by capturing various behavioral profiles (such as gait, keystroke, etc.). For increased assurance we employ Mobile Connect (MC)[4], which is the equivalent of a secure SIM authenticator where the Mobile Network Operator (MNO) act as an IdP. Therefore, promoting the device to the main authentication gateway not only eases the user from the burden of remembering multiple complex passwords, but also facilitates technically complex but needed authenticators that make our architecture fully aligned with the latest NIST standards for authentication[5].

However, we admit that the mobile device becoming the main authentication gateway is not by itself a universal remedy as it entails serious caveats. First, it becomes a single point of failure in case of device loss or failure; we believe that the lack of an efficient device failure/loss recovery mechanism is the main reason passwords are still in use and they have not been replaced by RSA keys. Second, the device is vulnerable to hijacking after the user has been authenticated. To overcome these problems, we propose a reliable failure recovery mechanism by leveraging a centralized entity, dubbed Identity Consolidator (IDC), in conjunction with MC authentication and a separate entity for behavioral authentication, called Behavioral Authentication Authority (BAA). At the same time, BAA ensures that unauthorized access to services by illegitimate holders of the device is prevented. Besides failure recovery, the IDC also offers identity and privacy management and allows to prove combinations of fragmented identity aspects, thus solving the identity fragmentation problem.

Contributions. In summary, with the proposed architecture we make the following contributions:

1. We demonstrate the merits of the seamless integration between strong/usable password-less authentication methods and federated login solutions.

2. We offer support for privacy-preserving ABAC on the mobile device.

3. We propose the separation of concerns for Authentication, Authorization and Behavioral Authentication to IdPs, SPs and BAAs respectively. This enables the incremental deployability of the proposed architecture.

4. We provide a rich set of features to the user through the IDC. Specifically, a user can manage the spectrum of her online accounts and define options that will enhance her security, privacy and user experience on the Web.

5. We propose the use of an innovative failure recovery mechanism, which is realized through the IDC, and behavioral and MC authentication.

Organization. In Section 2, we define important terminology and the required background. In Section 3, we define the threat

[4]https://www.gsma.com/identity/mobile-connect
[5]NIST stands for National Institute of Standards and Technology (NIST) https://pages.nist.gov/800-63-3/

model and the requirements that enable the design of our architecture. Section 4 provides a description of the main components our proposed architecture comprises. In Section 5, we report the design of our architecture while in Section 6 and Section 7 we describe and evaluate our prototype implementation, respectively. Finally, we review the related work in Section 8, and we conclude in Section 9.

2 Terminology and Background

2.1 Terminology

User Device (UD). This is the main gateway to get to DCA. In this work, we assume a user device that is able to utilize recent advances in the field of Trusted Execution Environments [25]. This enables the device to securely safeguard cryptographic credentials within its software stack.

Identity Providers (IdP). These are trusted entities that are responsible for securely maintaining and transferring end-users' identity attributes. They incorporate strong authentication mechanisms so that they can regulate end-users access. In the context of Privacy-Preserving Attribute-based Access Control, IdPs are responsible for issuing and verifying the end-users' cryptographic credentials.

Identity Consolidator (IDC). This is a centralized trusted entity that acts as the main IdP and manages all the access control needs of the user. The user is able to authenticate to the IDC, issue and verify cryptographic credentials, perform failure recovery (in case of lost or damaged device), and lock/unlock its online accounts.

Service Providers (SP). These are entities that are responsible only for authorizing end-users to their service. All other critical operations (i.e., authentication, verification of credentials) are performed by delegating them to trusted entities (IdPs) via Federated solutions, such as OpenID Connect (OIDC).

Behavioral Authentication Authorities (BAA). BAAs are special instance of IdPs that offer behavioral authentication to SPs. These entities maintain various behavioral profiles for each user that are obtained using signals that are either captured by the user's device or by the BAA itself, depending on the trait type. For instance, a BAA deployed by a Telco is able to capture the browsing history or traffic patterns on its own, whereas a gait trait requires signals from the user device. Based on these profiles, BAAs offer on-demand and continuous behavioral authentication to SPs.

2.2 Background

Federated Authentication. SPs can delegate the authentication to a trusted entity (i.e., IdP) that can authenticate the end-user with strong authenticators without the need of modifications in the SP's authentication stack. Furthermore, federated login solutions are privacy-enhancing because user information is stored and maintained in secure IdPs and can be managed by the end-users. At the same time, the secure delivery of verified identity attributes to SPs is enabled. In this work, we propose the use of the OIDC specification, which works as follows: When an SP needs to authenticate an end-user, it redirects him to an IdP in order to authenticate her. After the authentication is completed at the IdP (through an IdP-defined authentication mechanism), the end-user is redirected back to the SP, which can identify who the user is. Then the SP can obtain, with the user's explicit consent, identity attributes from the IdP.

Strong and Usable Authentication Mechanisms (FIDO UAF). One of the main objectives of the proposed architecture is to offer a secure and usable authentication solution. To this end, we propose the use of a secure password-less solution that use strong cryptographic operations in order to authenticate end-users, namely FIDO UAF. FIDO UAF utilizes biometrics in order to locally authenticate end-user and strong cryptographic operations to authenticate the device with the remote service. When a user authenticates with her device using biometrics, he unlocks the stored cryptographic keys which are subsequently used for authentication to the remote service.

Privacy-Preserving Attribute-based Access Control (PABAC). We propose the integration of attribute-based state-of-the-art cryptographic credentials stacks with Federated solutions. Specifically, we propose the deployment of Idemix [8] and U-Prove cryptographic stacks on trusted entities (IdPs), and the use of OIDC to prove attributes to SPs in an anonymity-preserving fashion.

3 Threat Model and Requirements

In this section we define the threat model and the requirements for the proposed architecture. Both the requirements and threat model guide the design and definition of our architecture as described in Section 4 and Section 5.

3.1 Threat Model

The proposed architecture faces various threats that we must identify. We categorize the identified threats according to the main components of our architecture.

User Device. The mobile device of the user is the most vulnerable component in our architecture. We admit that the mobile device can be stolen by an attacker who might or might not be able to perform software (i.e., side channel attack) and/or hardware attacks.

Service and Identity Providers. Like every online service, the SPs in our architecture face various threats. First, we have to ensure that the tokens and all the messages exchanged between the server and the clients are protected and will not be disclosed to an attacker during an authentication. Second, we assume an attacker who is able to perform Active (Man-in-the-Middle (MitM), Impersonation, Session Hijacking), Cross Site Request Forgery (CSRF), and Replay attacks. Last, a compromised IdP is another threat.

User Privacy. The privacy of the user is of vital importance in our architecture. A malicious SP is in place to identify a user through a combination of context from a series of transactions. Even if standard anonymization practices are performed by the

user, if two or more authorized entities (SPs and/or IdPs) are colluding, the user can be identified.

3.2 Requirements

To provide a complete solution and address all the aforementioned problems and threats, our architecture should fulfill the following requirements:

R1: Standards Compliance. The proposed system should be compliant with open standards. This is crucial as it allows incremental deployability, which can lead to the wide adoption of the proposed architecture.

R2: Ease of deployment. Incremental deployability is important for the wide adoption of our solution. SPs participating in our architecture should be able to offer strong authentication mechanisms to their end-users without the need to modify their software stack.

R3: Identity Federation and Management. To combat identity fragmentation, users should have a federated identity on the web that they can use to prove various attributes of their identity to IdPs and/or SPs in order to get access to specific resources. This requires a centralized entity that will consolidate the various online accounts of a user while enabling him to maintain control over her identity attributes.

R4: Failure Recovery. All user access control needs should be anchored to her device, which enables authentication with various usable and cryptographically strong methods. Furthermore, the proposed architecture should support appropriate failure recovery mechanisms in case of device loss, theft or failure. This will allow the unobstructed access to online services during unfortunate events.

R5: Privacy-preserving ABAC. In this work, we aim at providing attribute-based authentication while preserving users' privacy. In a typical ABAC scenario the SPs should run the appropriate cryptographic verification stacks in order to be able to authenticate specific attributes. However, this introduces deployability issues since not all SPs are able to run exotic cryptographic stacks. Thus, a critical requirement is to enable SPs that do not run cryptographic credentials to support privacy-preserving ABAC.

R6: Multi-factor Authentication. SPs that provide access to critical resources may require additional authentication for their users for higher assurance. Because of that, our architecture should offer additional authentication mechanisms to be triggered whenever SPs wish to further verify the identity of a user.

4 Architectural Overview

In this section we describe the main pillars of our architecture. This architecture consists of the following: 1) User Device; 2) Identity Consolidator; 3) Identity Provider; 4) Service Provider; and 5) Behavioral Authentication Authorities. Fig. 2 depicts our proposed architecture including its main components and the interfaces that interconnects them. All the communications between the components are built around OIDC protocol and by switching SP and IdP roles. Below we describe in detail the functionality and the modules that comprise each component.

4.1 User Device (UD)

The mobile device of the user is central in our architecture as we aim to provide device-centric authentication. We take advantage of the FIDO UAF protocol to make the user's device the main gateway for accessing services on the web. By deploying the FIDO UAF protocol stack we enable human-to-device authentication using biometrics (e.g., fingerprint). The device also runs federated authentication protocols (such as OIDC) with IdPs and SPs (aka, relying parties) for authorization and authentication purposes.

Furthermore, we deploy cryptographic credential stacks (Idemix and U-Prove) on the device to enable PABAC. These stacks allow users to request from the IDC and/or their IdPs the issuance of cryptographic credentials. The issued credentials are stored in a secure fashion in the Cryptographic Credentials Storage (CCS), which is also part of the user's device. The cryptographic credentials stack is also responsible for revealing issued credentials to IdPs during an authentication. Credentials stored in the CCS should not be exported even if the device has been compromised. This is achieved using a Trusted Execution Environment.

To enable continuous and second-factor authentication, the software running on the mobile device includes a behavioral profile capture module, which is responsible for capturing the behavior of the user taking advantage of the various sensors available on the mobile device.

4.2 Identity Consolidator (IDC)

The IDC is an integral component in our architecture. It is a centralized fully trusted entity that can be considered as a special instance of an IdP, which offers identity and privacy management and is required for failure recovery. The IDC collects identity attributes from various IdPs upon a user's request. The collected attributes are securely stored in a repository within the IDC. The following modules comprise the IDC: a) Authentication management; b) Account management; c) Identity and Consent management; d) Credential management; and e) Identity integration.

Authentication Management Module (AuthMM). It encapsulates a FIDO-enhanced federated login protocol, which allows the IDC to act as an OpenID Connect IdP for undertaking FIDO authentication. This module also allows the IDC to run federated login protocols for transferring identity attributes between distinct IdPs. Apart from these, the AuthMM also offers the appropriate failure recovery mechanisms in cases where the user loses access to her device.

Account Management Module (AMM). This module enables the users to manage the status of their accounts in various SPs and IdPs. A user can protect her accounts by locking access to them in case of device loss. The IDC can also act on behalf of the user and lock her online accounts when it detects a high risk of account compromise. The AMM is responsible to keep track of all the BAAs, SPs, and IdPs of a user and it also allows BAA,

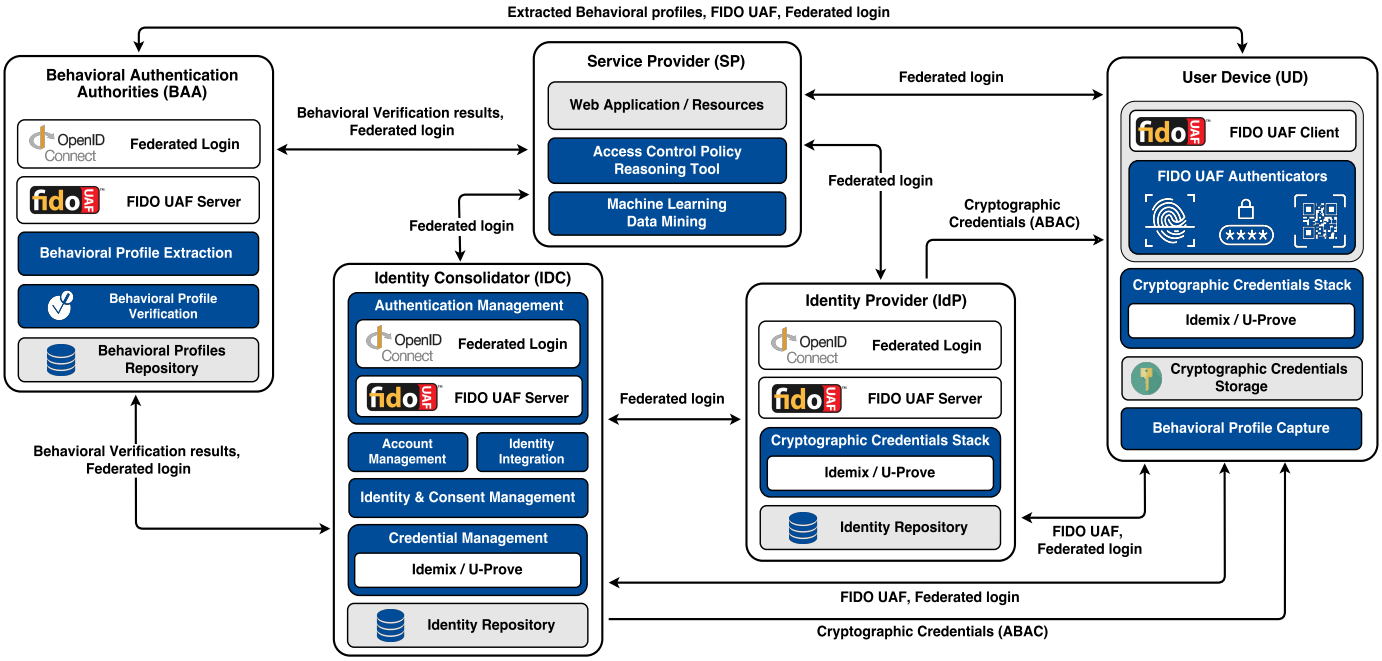

Figure 2: Privacy preserving architecture for device-centric and attribute-based authentication. The main architectural components with the modules that comprises each component.

SP, and IdP admins to register their entities with the IDC. Using this knowledge, the AMM acts as a BAA discovery service for SPs that may require a second-factor authentication. Besides these, the AMM allows the users to manage their IDC account, for example to set their preferred degree of privacy within the IDC (e.g., no attribute stored on the IDC) or completely delete her account. Lastly, the AMM facilitates the integration of the Mobile Connect protocol within our architecture. To achieve this, IDC act as a relay for SPs that request MC authentication (see Subsection 5.4). This is important because SP registration to Mobile Connect is costly.

Credential Management Module (CMM). The CMM enables ABAC in our architecture. This module runs cryptographic credential stacks (Idemix/U-Prove) that allows users to issue cryptographic credentials, from their verified identity attributes, directly to their mobile device and then use them to access a variety of SPs. The issued credentials can also be backed-up at the IDC for failure recovery purposes. The CMM also provides the required functionality for managing cryptographic credentials.

Identity Management Module (IMM). IMM empowers users to manage their identity information. This service consists of the profile and the consent management modules. The profile management provides easy browsing and management of the identity attributes that IdPs and SPs know about a user and informs her about the risks of involuntary attributes inference. It also allows users to transfer attribute values between different IdPs by extending federated login protocols (OpenID Connect). The consent management allows users and IdPs to define consent policies with respect to revealing specific attributes to specified SPs and IdPs.

Identity Integration module (IIM). The main responsibility of this module is the standardization and normalization of the users' identity information. We acquire the users' identity information via physical means (e.g., leveraging the Near Field Communication (NFC) technology[6] to read the user's e-Passport information) and we also perform online identity acquisition where the IDC acts as an SP to receive the users' identity attributes from other IdPs through OIDC. The IIM encapsulates the required logic for combining, fusing, inferring and validating identity attributes.

4.3 Identity Providers (IdP)

Within our architecture, IdPs are entities that authenticate users and share their identity attributes with SPs. Each IdP has an identity repository that stores users' attributes. IdPs also run cryptographic credential stacks (i.e., Idemix and U-Prove) that facilitate the issuance or verification of cryptographic (PABAC) credentials from the stored identity attributes.

4.4 Service Providers (SP)

SPs require minimal modifications. Namely, they only have to run an OIDC client in order to communicate with other entities in our architecture. SPs are also able to support FIDO and PABAC without the need to run any sophisticated cryptographic stacks by involving IdPs in the authentication process. Furthermore, SPs incorporate their business logic within access control policies. Access control (AC) policies can be managed by the SP administrator using an Access Control Policy Reasoning tool. Taking into account the defined AC policies, this tool evaluates users' requests on resources based on their provided attributes. Besides this, the tool also recommends to SPs' admins policy improvements derived from a machine learning

[6] http://en.wikipedia.org/wiki/Near_field_communication

algorithm that considers existing policies and the types of requests.

4.5 Behavioral Authentication Authorities (BAA)

BAAs are separate entities that provide both on-demand and continuous behavioral authentication as part of an entire DCA solution. To achieve this, BAAs continuously track the users' behavior through various means and offer a behavioral solution to either SPs or the IDC as a second or third factor authentication. Specifically, when requested by an SP or the IDC, BAAs act as an IdP that can verify whether the behavior of a user remains consistent with her usual habits. The behavioral authentication outcome is released to the aforementioned entities using OIDC. However, since behavior is not privacy-preserving we offer behavioral authentication as a second or third factor while the first-factor can be privacy-preserving using PABAC.

4.6 Privacy-Preserving Attribute-based Access Control (PABAC)

We enable PABAC by integrating the Idemix and U-Prove cryptographic credential stacks within the OIDC Provider on the IdPs. In this way, an IdP can act as a credential issuer and/or verifier. Users can request the issuance of cryptographic credentials by these IdPs or the IDC. This solution has various advantages which are: 1) SPs are not required to deploy any cryptographic credential stacks to support PABAC. Instead, they delegate the verification of PABAC credentials to IdPs; and 2) it allows for more flexibility as PABAC-enabled IdPs might not be collocated with SPs.

5 Design

In this section we provide adequate information regarding the design of our architecture and how we address the requirements set above.

We propose an architecture in which everything is built on top of the OIDC specification. We choose to use OIDC with infrastructure authenticator IdPs for incremental deployability. This is a central design choice that allows us to clearly separate the concerns of SPs and IdPs during an authentication, thus addressing requirements R1 and R2.

5.1 Diverse Authentication Framework

We propose a NIST-compliant [1] diverse authentication framework for the end-users. Specifically, our federated architecture offers various authentication modalities, thus supporting all the assurance levels defined by NIST. Depending on which is used, the granted Authenticator Assurance Level (AAL) is determined. For example, the highest degree of assurance (AAL3) requires a hardware-based cryptographic authenticator and two-factor authentication. We achieve this with an enhanced FIDO UAF specification that takes advantage of the TEE that run on end-user devices combined with a secure SIM (Mobile Connect). Here we assume that in the future FIDO and Mobile Connect will be as secure as a hardware cryptographic token (FIPS 140-2[7]) because of advances in TEE.

Moreover, a backup password along with behavioral authentication provides the lower degree of assurance (AAL1), while FIDO UAF authentication alone provides AAL2.

5.2 FIDO-enhanced Federated Authentication

OpenID Connect is a simple federated identity layer on top of the OAuth 2.0 protocol[8], which facilitates federated authentication. Thus, OIDC specification enables SPs to delegate the authentication of end-users to IdPs, as well as to obtain profile information about an end-user from the IdPs in an interoperable manner.

The FIDO Alliance provides the FIDO UAF specification, which is a password-less solution that enables IdPs to authenticate end-users using strong authenticators (e.g., fingerprint) for user-to-device authentication and cryptographic protocols for device-to-service authentication (e.g., RSA). By combining the concepts of strong authentication alongside with the delegation of authentication to IdPs we allow for a more user-friendly and secure solution for end-users.

5.3 Federated Privacy-preserving Attribute-based Authentication

The various components that comprise our architecture were carefully designed in order to provide a PABAC solution on top of the OIDC while also addressing requirement R5. PABAC enables SPs that are not aware of any cryptographic credentials stack to allow end-users to use cryptographic credentials and get access to their resources. To this end, we propose a custom authentication module within the OIDC Provider that acts as Idemix/U-Prove verifier, thus allowing IdPs to issue and verify cryptographic credentials. In fact, with this module we modify the OIDC Provider so that it uses one-time pseudonyms instead of persistent unique identifiers.

Federated PABAC offers two concepts of anonymity, namely untraceability and unlinkability. Untraceability is the security property that precludes the IdP that issued an attribute credential from tracking to which SP the credential has been shown. Unlinkability is the property that prevents an IdP or SP from realizing that two or more distinct sessions under the same attribute credential have been initiated by the same user [8]. At the same time, users' privacy is preserved since they are able to authenticate to SPs by revealing only the required attributes without revealing their complete identities.

5.4 Mobile Connect (MC) as a Service

In our architecture we enable SPs to authenticate users using MC. Though the IDC we offer MC as a Service, thus allowing incremental deployability of the MC protocol even if the SP is not registered with the MC API Providers. To achieve this, the IDC acts as a proxy to SPs for discovering and contacting MC IdPs (Mobile Network Operators-MNOs) on behalf of the

[7] https://csrc.nist.gov/publications/detail/fips/140/2/final
[8] https://oauth.net/2/

SP. In this way, we see MNOs as any other IdP within our architecture using OIDC. The IDC acts as an MC SP to retrieve the required attributes and then acts as a vanilla OIDC provider that proves those attributes to another SP that is not registered in the MC ecosystem. In this way, the SPs do not have to be aware of the MC protocol. They just need to know the value of attributes that can be verified at the required AAL only by MC IdPs. Which attributes are those and how they can be retrieved is knowledge that is available only to the IDC.

5.5 Failure Recovery Framework

When moving the authentication to the mobile device there are serious caveats that we should consider as we also underline in requirement R4. The most crucial one involves recovery after device loss or failure. Another problem is that in case the device is stolen, the thief has direct access to the secret. We address the first problem via the IDC that federates multiple independent factors (e.g. MC and BAA). These independent factors can be easily used in conjunction with a single secure backup password or physical identity verification to reliably authenticate the user during recovery. The second problem is addressed via FIDO on devices.

Using MC and a BAA the user has to first login to the IDC using her secure backup password, which is required only for failure recovery. By doing so, she is granted only temporary and tentative access (AAL1), which provides limited functionality. In particular, she cannot view, restore or manage credentials and identity attributes. Subsequently, the IDC acts as SP authenticating the user through a Telco IdP via MC. Because the user cannot use FIDO to authenticate, she is able to authenticate via SMS[9] using her newly issued by the MNO SIM card. In case of device theft or loss, to ensure that the authentication attempt is performed by the legitimate user, the IDC needs to confirm with the MC IdP that the given device was reported as lost and a new SIM card was issued. Additionally, in case the user is not registered with MC then she can use any other OIDC/FIDO-compliant IdP.

For increased assurance the IDC also needs to authenticate the user via one of the trusted BAAs that are registered under her account. The user authenticates to her BAA using a backup password (specific to the BAA). We note that the user does not have to memorize this backup password since she is able to backup all her backup passwords to the IDC. BAAs can have insecure and easy to memorize backup passwords as their authentication modality is behavioral and the backup password is used only to prevent denial of service attacks.

After the user has authenticated, the BAA grants the user tentative access and she is not allowed to manage her behavioral profile until her signature is verified as that of the legitimate user's. With the user having tentative BAA access, the device sends behavioral records to the BAA, while all the records prior to the new device login are not considered for the authentication. The IDC acts as an SP while the BAA acts as an IdP authenticating the user based on her behavior. Once the BAA has collected sufficient records to give a verdict on whether the user behaves as usual the result is returned to the IDC via OIDC. If the verdict is negative the BAA locks that device out of its IdP. If the verdict is positive, then the user is granted full access (AAL3) to the IDC and the BAA issues new FIDO credentials for her account to the new device. Both MC and BAA authentication is needed because BAA does not formally increase the NIST authenticator assurance level but it is just an extra assurance.

We note that if the user does not wish to use a backup password, she is able to recover from failure with physical identity verification by scanning her ePassport using her mobile device. Leveraging the NFC capabilities of the device we are able to acquire the verified identity of the user. If the acquired identity matches the one that she had proved to the IDC before the failure, then she is granted temporary and tentative access to the IDC.

5.6 Multi-device Support

For usability purposes, we identified the need to support multiple devices. Thus, we modified the FIDO UAF client and server software so that it allows the user to register multiple FIDO cryptographic keys, one for each device they use, for each account they have. This modification enables the users to maintain multiple devices.

Besides this, we also support multiple type of devices. For example, a user is able authenticate to an SP through her desktop computer. To achieve this, we integrate a Quick Response (QR) authentication server within the IdP's OIDC software to enable authentication from desktop computers to SPs using FIDO. Therefore, there is no need for the users to run any user device components on their desktop computers.

We acknowledge that the availability of the mobile device of the user is crucial since a mobile device is required for authentication. However, this is also a limitation for FIDO and DCA in general.

5.7 De-anonymization Risks and Privacy Assessment

Preserving users' privacy is of vital importance in our architecture. Thus, we provide to the users privacy risk indicators that define the risk of involuntary de-anonymization. To achieve this, we extend OIDC so that it keeps logs of the identity attributes revealed to SPs. De-anonymization risk calculation can be separated into two categories based on the protocol that a user is using to authenticate. The first concerns the de-anonymization risk calculation for vanilla OIDC whereas the latter for PABAC (Idemix/U-Prove).

In the OIDC case, we calculate the confidence probability of whether an SP can infer the value of an attribute that the user has not explicitly revealed based on which attributes she has already revealed. Due to their nature, Idemix and U-Prove provide unlinkability and untraceability. This differentiates the risk calculation from the one performed for OIDC. This calculation does not depend on the attributes that the user has shared with

[9] We acknowledge the vulnerabilities of the SS7-based SMS system [3]. In any case the authentication to the MNO IdP can take place in secure ways like FIDO where the public key of the device is installed during the new SIM registration or with a secure version of SMS [19]

an SP in the past since PABAC prevents the SP from linking new sessions with past ones. The calculation is made based on the attribute or combination of attributes that the user is about to share with an SP.

Note that if the user uses untraceable and unlinkable attribute-based authentication the de-anonymization risk depends on the rarity of the attribute combinations presented to the IdP and SP in a given population and the degree that the SP and IdP know the distribution of attributes in the population.

5.8 Deployability and Adoption

Federated architectures have many significant benefits for adopters. First, user experience is enhanced since an end-user has to consolidate and prove her identity once at the IDC and then it can be reused to access multiple IdPs and SPs. Second, their is a significant cost reduction to both the end-users (reduction in authenticators) and the SPs (reduction in infrastructure).

On the one hand, end-users do not have to remember dozens of passwords and at the same time they are able to retain their anonymity using PABAC. On the other hand, SPs can offer FIDO and PABAC authentication to their end-users without the need to deploy any cryptographic stacks within their infrastructure. In addition, there is significant data minimization for SPs because they do not need to pay for collection and storage of personal identity information. As a result, SPs can focus on their mission rather than the business of identity management [1].

Furthermore, it is clear that IdPs are crucial in federated architectures. However, what are the incentives for an organization to play the role of the IdP? By participating in our architecture, an IdP has many benefits. For example, an organization who maintains identity information about users (such as age) can offer age verification services to SPs who require age verification from their end-users in order to abide by the online age verification requirements imposed by regulators like the Gambling Act 2005 legislation[10] for remote gambling in UK.

6 Implementation

In this section we provide the details of our prototype implementation. We implemented all the architecture components as well as all the protocol extensions and integrations that we describe in Section 5.

OIDC/FIDO UAF: To exploit the OIDC Provider features, we make use of the OpenAM software[11]. We implemented, within the OIDC Provider, a custom authentication module, called FIDO UAF authentication module, which is responsible for undertaking the authentication of the users according to the FIDO UAF specification. To achieve this, our custom authentication module communicates with the FIDO UAF Server using a REST interface. The FIDO UAF server handles the authentication of the user by communicating with the FIDO UAF client that runs on users' devices.

OIDC/PABAC: PABAC is realized through the deployment of Idemix/U-Prove credential stacks. To enable IdPs to act as credentials issuers/verifiers we have implemented a custom authentication module within the OIDC Provider. For this purpose we use the FIWARE REST API[12], which is able to utilize both underlying cryptographic protocol stacks used in our architecture.

Identity Consolidator: We have implemented the IDC component and its respective modules as a web application. Within the IDC we implemented a well defined REST interface that allows all the other components of our architecture as well as all the external entities to interact with the IDC.

Moreover, we also implemented an MC proxy service. More precisely, we implemented a custom module within the IDC that allows the IDC to act as an MC proxy. This custom module invokes a GSMA Apigee API Exchange-enabled[13] discovery service on a trusted MC Provider. This API is mainly used as the federation mechanism for MC authentication.

Behavioral Authentication: Stand-alone BAA entities offering on-demand and continuous behavioral authentication is one of the most important contributions in this work. In the context of this work, we implemented a prototype BAA entity for gait verification including its client-side modules that capture the gait of the end-user on her device. The captured behavior data is immediately transferred to the BAA server through an authenticated secure channel, and once the transmission is completed we securely erase the behavioral data from the mobile device.

User Device: We implemented an Android application that incorporates all the required user device functionality. This application runs a FIDO UAF client, the behavior capturing module, and utilizes the TEE to store cryptographic credentials. We increase maintainability by implementing each module as a separate Android library.

7 Evaluation

In this section we evaluate our prototype implementation in terms of performance, User Experience (UX), and security.

7.1 Performance Evaluation

Here we assess the performance of the proposed authentication solution (both OIDC/FIDO and OIDC/PABAC) against the performance of the vanilla OIDC, FIDO UAF, and Idemix protocols. Each experiment was conducted by sending a batch of authentication requests within a second starting from 500 to 4000 requests, while measuring the average response time of the server for each batch of authentication requests, this being the time for all the authentication messages to be exchanged between the client and the server. We note that all the authentication requests were successful.

OIDC/FIDO UAF. As described in Section 6, we implemented a custom authentication module by deploying a FIDO UAF server to the IdP's software stack to enable IdPs authenticate end-users using FIDO. Here, we evaluate this deployment in

[10] https://www.legislation.gov.uk/ukpga/2005/19/contents
[11] https://forgerock.org/openam/
[12] https://goo.gl/dkG5R8
[13] https://apigee.com/about/tags/api-exchange

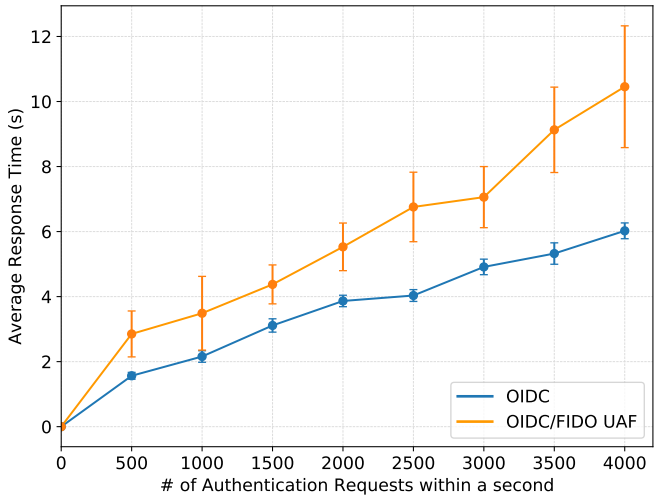

Figure 3: Average response time of an OIDC/FIDO UAF authentication request and vanilla OIDC authentication request.

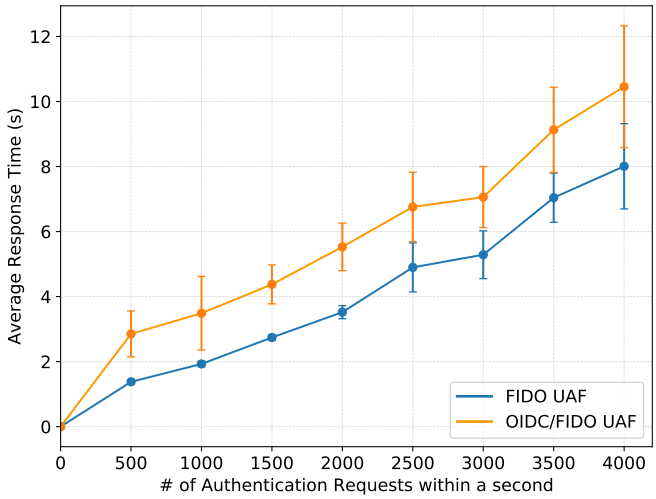

Figure 4: Average response time of an OIDC/FIDO UAF authentication request and vanilla FIDO UAF authentication request.

terms of performance to identify: 1) how it performs under high authentication demands; and 2) the overhead that our custom authentication module introduces compared to the vanilla OIDC and FIDO protocols. We note that in our measurements we do not count any user-induced delays (i.e., the time the user needs to enter her password) and we use a 20 characters long password for authentication wherever this is required (vanilla OIDC).

Initially, we evaluate the performance of a vanilla OIDC deployment by employing the standard OIDC authentication process. A similar evaluation was also conducted for the vanilla FIDO UAF authentication process. All the simulations were performed by porting an Android client on a desktop, which implements the required functionality for standard OIDC authentication as well as for FIDO authentication, and we simulate the parallel authentication processes using different threads.

Following the same approach, we evaluate our OIDC/FIDO UAF authentication module. Again, the simulations were performed by running our Android OIDC/FIDO client implementation on a desktop. We repeat each experiment 10 times and we calculate the 95% confidence interval of the average response time of each deployment as the number of authentication requests increases. Figures 3 and 4 present the results of the evaluation of our custom authentication module as well as those of vanilla OIDC and FIDO. We observe that our custom authentication module scales along with the number of authentication requests, with the server's average response time not being drastically impacted. Compared with the vanilla OIDC and FIDO, our implementation does not introduce any substantial delay in the authentication process: when the number of simultaneous authentication processes is 4K, the average response time of an OIDC/FIDO authentication request is 4.5 sec and 2.4 sec more than the average response time of the vanilla OIDC and FIDO protocols, respectively. Considering the advantages of our proposed solution, we consider the additional delays as negligible.

All the experiments were conducted using an OpenAM software stack and a FIDO UAF server which run in separate docker[14] containers on the IdP and the client requests are executed using an Internet connection.

OIDC/PABAC. Next, we conducted a similar evaluation of our OIDC/PABAC custom authentication module. For this evaluation we implemented a custom OIDC/PABAC authentication module by utilizing the Idemix credentials stack. We deployed the implemented module to the IdP's software stack to enable the IdP to act as an Idemix credential issuer and verifier. The purpose of this evaluation is to identify: 1) how our PABAC-enabled IdP performs under high authentication demands; and 2) the overhead that our implementation introduces compared to the vanilla OIDC and Idemix protocols. We note that in our measurements we do not count the time required for the issuance of the Idemix credential.

We evaluate both a vanilla Idemix deployment and our OIDC/PABAC implementation. Similar to the OIDC/FIDO evaluation, we repeat each experiment 10 times and we calculate the 95% confidence interval of the average response time of each deployment as the number of authentication requests increases. Figures 5 and 6 present the results of the evaluation of our OIDC/PABAC implementation compared with those of the vanilla Idemix and OIDC protocols, respectively.

We observe that the average response time of our custom authentication module follows a similar trend to the one of vanilla Idemix authentication and it does not introduce substantial delay in the authentication process: when the number of parallel authentication processes is 4K, the average response time of an OIDC/PABAC authentication request is 5.3 sec more than the average response time of a vanilla Idemix authentication request.

7.2 User Experience (UX)

It is widely accepted that the quality of the User Experience (UX) determines the success or the failure of any new solution.

[14] https://www.docker.com/

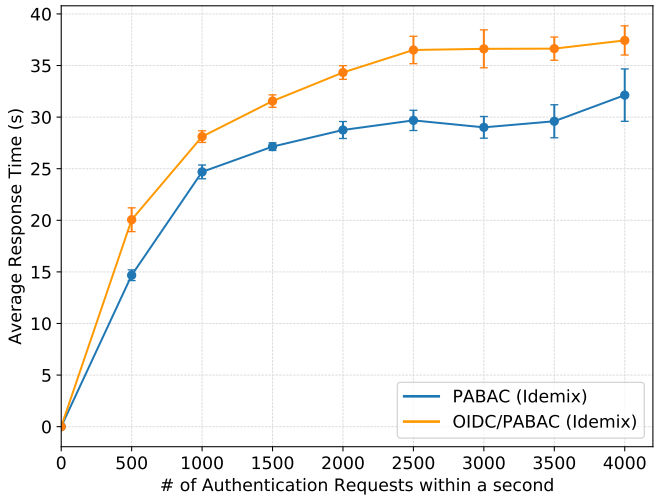

Figure 5: Average response time of an OIDC/PABAC authentication request and vanilla Idemix authentication request.

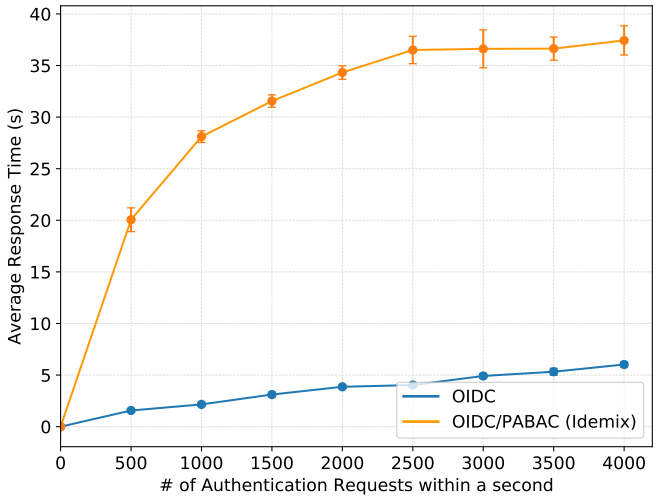

Figure 6: Average response time of an OIDC/PABAC authentication request and vanilla OIDC authentication request.

Heuristics	**Positive (%)**	**Negative (%)**
Easiness	90	10
System Status Visibility	88	12
Language	92	8
Control & Error Prevention	93	7

Table 1: Usability heuristics and their respective percentage of positive/negative evaluation.

The UX includes "all the aspects of how people use a product: the way it feels in their hands, how well they understand how it works, how they feel about it while they are using it, how well it serves their needs, and how well it fits into the entire context in which they are using it" [5]. For this reason, the proposed solution needs to be assessed not only in terms of usability, but also considering the way it enhances the whole UX. The UX evaluation aims at ensuring that the solution is perceived as useful and desirable according to users' needs, easy to learn, as well as effective and efficient to achieve its specified goals (i.e. efficient failure recovery).

The evaluation is performed through an iterative process in which each step provides recommendations to further improve the UX: 1) preliminary assessment by experts, who analyze the level of compliance with the main usability heuristics [27] and identify usability issues to fix; 2) test with a sample of end-users through the "think aloud method", so to collect empirical data while observing the users interacting with the system to perform realistic tasks [31]; and 3) collection of users' feedback through an online questionnaire. The focus of such evaluation is not only on tasks and operations within the workflow, but also on users' expectations and perceptions related to the design concept, the features, the information architecture, as well as the aesthetics.

Table 1 presents the preliminary results collected from 41 respondents through the online questionnaire. The participants, both male and female (aged 20-24 years), were recruited from the Computer Engineering Department of Cyprus University of Technology. The results of the survey provide positive feedback supporting the design, and suggestions to improve the UX. Considering the impact on the UX, most of the respondents (90%) agree that the solution provides a reliable and secure authentication mechanism, and it makes the access to web services easy and quick. Most of the respondents (92%) consider the language (i.e. terms, messages) familiar and comprehensible and the organization of the information clear Furthermore, 88% of the respondents believe that the system keeps the users informed about what is going on through the appropriate feedback (system status visibility), while 93% of them agree that the system enables the user to easily diagnose and bypass errors. While the weak aspects highlighted by the respondents are related to learnability, help and documentation as well as efficiency of the tasks and navigation flow. Thus, the further improvement of the solution will address such issues.

7.3 Security Analysis

Here we discuss how we defend against all the possible attacks defined in our threat model. We categorize the threats and the mitigation strategies that we employ to defend against them according to the main components of our architecture.

User Device. The mobile device of the user is the most critical and vulnerable component in our architecture because it can be stolen. First, assuming that we have an attacker who has stolen the device and is not able to perform software attacks, our architecture is able to effectively defend against such a threat by employing multi-factor authenticators that need to be activated through a biometric (FIDO and continuous behavioral authentication) that can prevent an attacker from being authenticated as the legitimate user; and more importantly by offering a specialized account locking module that is part of the AMM of the IDC allowing the user to lock access to her online accounts on the stolen device.

Next, we also examine the case where the attacker is able to perform software attacks. If the behavior capturing proto-

cols run in the Normal World (Rich OS), a skilled software attacker can intervene and modify the contents of the memory while also modify the information captured from the device's sensors. However, since all the local measurements are immediately sent to the BAA and are not stored locally on the device then we can prevent such an attacker from bypassing the behavioral authentication. On the other hand, if the protocols run in the Secure World (aka Trust-Zone, or Trusted Execution Environment-TEE), no software attacker can compromise the memory and information paths. However, we do not have the ability to develop protocols for TEE as the trusted computing base has to be approved by vendors, such as Intel, Samsung, etc. We can just invoke specific services of it, such as storing cryptographic keys and performing secure cryptographic operations. For example, the activation of the on-demand behavioral authentication with a verifier is triggered through TEE-enabled secure biometrics (e.g., fingerprint). This is supplied by FIDO and can protect the user in case the device has been recently stolen and the behavioral signature has yet to change.

Last, when an attacker is able to perform software and hardware attacks then she can bypass the trusted execution and present himself as the legitimate user only if the device is recently stolen, but again the owner of the device can lock access to her online accounts on that device using the AMM.

Service and Identity Providers. SPs and IdPs in our architecture face various threats. Initially, our solution guarantees that the tokens are never exposed to unauthorized parties during an authentication by establishing protected sessions between the SPs/IdPs and the users. All the messages exchanged between the SPs/IdPs and the users are sent over the Transport Layer Security (TLS) protocol.

Moreover, our architecture is able to defend against other types of attacks. First, to defend against Active attacks like MitM, Impersonation, and Session Hijacking attacks we generate access tokens to the authenticated users that are user- and scope- restricted. Second, to defend against CSRF attacks we perform header checks to verify the origin of the source and destination for every request while also using CSRF tokens in the communication between the user and the SP. Third, using TLS for all the communications between the user device and the IdPs/SPs we are able to defend against Replay attacks. Last, a compromised IdP is not considered since this is a general problem of federated architectures. If an IdP is compromised, it affects the authentication security only of the SPs that relies on that IdP.

User Privacy. Two or more authorized entities (IdPs and SPs) acting maliciously are considered attackers and might be in place to identify a user. However, we are able to preserve the privacy of users by employing advanced unlinkable and untraceable cryptographic credentials that are used by the users to authenticate with PABAC-enabled IdPs. Additionally, using the Consent Management module of the IDC, a user is able to provide her consent when revealing identity attributes to SPs. Last, the Profile Management module of the IDC offers to the users useful privacy risk indicators for each one of their identity attributes. These indicators define the risk of involuntary de-anonymization as well as the possibility of an attribute inference.

8 Related Work

In this section we review existing work on password paradigm alternatives, behavioral authentication, identity federation and management, and attribute-based access control.

8.1 Password alternatives

It is evident that a 20 character random password used for a specific account is quite secure, albeit not user-friendly. Our ability to memorize secure passwords cannot compete with a computer's ability to guess them, thus it is impossible for users to memorize multiple complex and long enough passwords for each service they use.

In the last few years, the research community realized that the password paradigm is not an ideal solution able to cope with user authentication needs on the Web; mainly because of usability and security concerns. At the same time, even relatively secure passwords are not replay-resistant authenticators. Therefore, various studies aim at either replacing the password paradigm or propose solutions that mitigate its caveats. Specifically, [4, 26, 36] analyze the usability and security problems of the password paradigm. All studies pinpoint the password overload problem which leads users to choose easy to remember passwords or choose to reuse the same password across multiple domains. Also, users' perceptions of security seems to be an important factor that influences effective password usage.

To overcome these issues, password managers like PwdHash [30], LastPass [24], and RoboForm [2] allow users to use a variety of strong passwords for accessing their online services, while the burden of maintaining and remembering the password is offloaded to the password manager. However, some works [12, 37] highlight that the use of password managers introduce new security and usability issues. Namely, end-users cannot properly use password managers and this makes them susceptible to various attacks, while the protection mechanisms of several password managers have many security flaws. For example, most password managers are protected with a master password. If the master password is leaked to an adversary then the password manager becomes a central location for accessing the user's entire online presence. In contrast, in our solution a backup password is only required for failure recovery and not every time the user wants to authenticate with a service.

Additionally, passwords managers are susceptible to replay or server breach attacks, while in our solution even if an adversary overhears the challenge-response communication with the IdP, he cannot sign another challenge without the FIDO secure private-key. In case of a breach attack the compromised IdP only contains a perfectly useless list of public-keys. In addition, each private-key can be as long and random as needed to stay secure.

On top of the aforementioned, Karole et al. [23] highlight additional users' concerns with regard to the use of password managers. That is, they feel more confident in storing their passwords locally instead of cloud-based password managers.

Other studies propose alternatives to the password paradigm. Stajano [34] proposes Pico, a password replacement which relies on hardware tokens. At manufacturing time, SPs inject unique keys in each token, which are used for authentication purposes. In contrast to Pico, our architecture does not introduce any extra hardware. Instead, we leverage a user's mobile device to store the necessary cryptographic keys for authentication purposes.

Trusona[15] is a product that offers device-centric password-less and multi-factor authentication through a mobile application. A user can register by scanning one of her identity documents. Trusona is suitable for various use-cases ranging from online authentication to wire transfers. With the IDC, we go beyond Trusona offering sophisticated identity federation, proofing algorithms and standards-compliant FIDO authentication, as well as continuous behavioral authentication.

8.2 Behavioral Authentication

Behavioral Authentication provides an extra layer of security above our first factor of authentication. Seminal studies have shown that common security authentication mechanisms like PINs or patterns can be enhanced by adding the behavioral factor as another mean of authentication [38, 15]. Google has also identified the need for replacing the password with stronger and more usable authentication mechanisms. Thus, Google provides the Trust API[16], which offers multi-modal continuous authentication by tracking a user's behavior on her device (e.g., typing pattern, location, etc.).

Also, other solutions exist that continuously track users' behavior for authentication purposes based on various behavior types [22, 16, 33, 21]. However, the classifier's location in most of these approaches is not specified and they do not consider battery, computational and space limitations. At the same time, they only tackle observation and impersonation attacks. Unlike the aforementioned solutions, we propose an open architecture under which any entity able to capture user behavior can offer behavioral authentication via OIDC, while also offering enhanced protection against attackers that manage to compromise the device. In addition, we perform continuous behaviour authentication that allow us to have account- and device-wide lockdown when the device is not held by its legitimate user.

Chow et al. [13] propose TrustCube, a framework that leverages federated authentication schemes to authenticate users based on their behavior on behalf of any SP. Similarly, in our architecture, BAAs run OIDC for authenticating users based on their behavior. However, we go beyond TrustCube by offering BAAs as part of an entire DCA and identity consolidation solution.

NuData Security[17] and BehavioSec[18] offer continuous behavioral authentication software as a service. They use real-time behavioral and statistical analysis tools to resist attacks like account fraud, sharing, and takeover. These solutions are typically deployed on the SP and are application-domain-specific. In our approach, BAAs are independent entities that can employ any type of behavioral authentication. BAAs harvest user behavior data from an end-user's device in a non-intrusive and battery efficient way. Thus, they can provide via OIDC any type of indicator an SP deems necessary, spanning from a simple boolean flag to statistical scores.

8.3 Identity Federation and Management

The past two decades numerous identity federation and management solutions have emerged. One of them is the WSO2 Identity Server[19], which is an open source technology that when integrated within an SP's infrastructure can offer singe sign-on (SSO), and identity federation and management. Unlike WSO2, SPs in our architecture can have the same benefits by just running an OIDC client instead of having to deploy the whole solution into their infrastructure.

OpenID 2.0 [29] is a user-centric identity management platform in which each account has Identifiers (URI) at one or multiple IdPs, and enables an end-user to prove the possession of such an identifier. Users that own the accounts must remember each of their URIs, so some of them are used to access several SPs for validation and authentication of the user. If these SPs are malicious, then the users' attributes could be correlated and reveal their identities.

Other identity management approaches like Liberty[20] and SAML [9], offer federated user identities in a more privacy-preserving way. IdPs use pseudonyms or aliases to reference users to the SPs and these pseudonyms are different in each SP. One SP cannot directly reference a user in the namespace of another SP, thus preventing malicious SPs from colluding to correlate user identities. Inspired by this approach, we extend OIDC to employ pseudonyms so that user anonymity is maintained when they are used in conjunction with privacy-preserving cryptographic credential stacks on the IdP.

Venkatadri et al. [35] propose a framework that leverages information about identities that is aggregated across multiple domains to reason about their trustworthiness. The authors propose multiple ways for linking the multiple online identities of a user (e.g., using SSO protocols) that also enable the transfer of trust between domains without significant loss of privacy or implementation overhead for the IdPs. Instead, we deploy more sophisticated algorithms for assessing the trustworthiness of a user's identity with high confidence (see Subsection 4.2).

A more akin to our architecture solution is the Secure Identity Platform (SIP) by Civic[21]. Taking advantage of the blockchain technology, SIP offers a decentralized identity verification and management solution through a mobile application. Access to the identity information is protected with biometric authentication. We consider our architecture as a more concrete solution than SIP, offering a multi-factor password-less authentication experience, while at the same time PABAC preserves the privacy of the user.

[15] https://www.trusona.com
[16] https://thisdata.com/blog/androids-trust-api-a-short-history-and-why-its-a-game-changer
[17] https://nudatasecurity.com/
[18] https://www.behaviosec.com
[19] http://wso2.com/identity-and-access-management
[20] http://www.projectliberty.org
[21] https://www.civic.com/products/secure-identity-platform

8.4 Attribute-based Access Control

Attribute-based access control provides a boolean model in which resources are accessed only if the applicant has the appropriate access attributes as defined by the so-called policies. This access control model uses either one of two attribute based encryption (ABE) methods. Key-policy ABE [18] uses the policies to create the applicant keys and uses the attributes to describe the encrypted data. Ciphertext-policy ABE [6] uses a tree form access policy, where attributes are the leaves of the tree. Ruj et al. [32] propose a privacy-preserving access control scheme in the clouds, in which the attributes of each user belong to multiple key distribution centers [14]. The user's identity information is stored in the cloud and the cloud acts as the verifier for the users' credentials. However, user privacy is not protected in the cloud. Chase [10] introduces a multi-authority KP-ABE scheme that overcomes the drawbacks of a single authority attribute-based system. He proposes global identifiers to distinguish different decryptors and allows independent authorities to monitor attributes and secret keys in a distributed storage. Based on their first proposal, Chase and Chow [11] propose an improved version of the scheme were a polynomial number of independent authorities is set to monitor attributes and distribute secret keys.

In contrast with the above methods, we integrate in our architecture cryptographic credentials stacks (such as Idemix [8] and U-Prove) to let users prove their identity attributes to SPs using cryptographic credentials that are securely stored on their device. In addition, by integrating PABAC with OIDC we enable any SP to offer PABAC authentication without the need to deploy any cryptographic credential verification stacks.

9 Conclusions

In this work we propose an architecture for preserving privacy with device-centric and attribute-based authentication while also solving the serious caveats that the password paradigm has. It serves as an alternative for SPs that wish to replace their existing authentication mechanisms without the need to deploy any sophisticated software stacks. We readily admit that not all components of our architecture are individually novel. However, combining them together under one architecture, they produce the first proof-of-concept that password-less authentication can be done securely and in a user-friendly fashion under the device-centric paradigm. Our evaluation results show that our solution can be adopted by end-users and SPs without friction.

Acknowledgment

This research has been fully funded by the European Commission as part of the ReCRED project (Horizon H2020 Framework Program of the European Union under GA number 653417).

References

[1] Nist - digital identity guidelines. https://pages.nist.gov/800-63-3/.

[2] RoboForm Password Manager. https://www.roboform.com/.

[3] Ss7 sms-based exploit: A wake-up call to shift to stronger two-factor authentication. https://blog.easysol.net/ss7_sms_based_exploits/.

[4] A. Adams and M. A. Sasse. Users are not the enemy. *Communications of the ACM*.

[5] L. Alben. Defining the criteria for effective interaction design. *interactions*, 3(3):11–15, 1996.

[6] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE SP '17*.

[7] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *IEEE SP '12*.

[8] J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *ACM CCS '02*.

[9] S. Cantor, J. Kemp, R. Philpott, and E. Maler. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. Technical report.

[10] M. Chase. Multi-authority attribute based encryption.

[11] M. Chase and S. S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *ACM CCS '09*.

[12] S. Chiasson, P. C. van Oorschot, and R. Biddle. A usability study and critique of two password managers. In *USENIX Security Symposium*, 2006.

[13] R. Chow, M. Jakobsson, R. Masuoka, J. Molina, Y. Niu, E. Shi, and Z. Song. Authentication in the clouds: A framework and its application to mobile users. In *ACM CCS '10 Workshop*.

[14] P. D'Arco and D. R. Stinson. On unconditionally secure robust distributed key distribution centers. In *ASIACRYPT*, 2002.

[15] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: Implicit authentication based on touch screen patterns. In *SIGCHI '12*.

[16] T. Feng, Z. Liu, K. A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In *2012 IEEE HST*.

[17] D. Florencio and C. Herley. A large-scale study of web password habits. In *WWW'07*.

[18] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. *IACR*, 2006.

[19] GSMA. Introducing mobile connect the new standard in digital authentication. https://www.gsma.com/identity/mobile-connect.

[20] B. Ives, K. R. Walsh, and H. Schneider. The domino effect of password reuse. *Communications of the ACM*, 47(4):75–78, 2004.

[21] M. Jakobsson, E. Shi, P. Golle, and R. Chow. Implicit authentication for mobile devices. In *USENIX HotSec'09*.

[22] Z. Jorgensen and T. Yu. On mouse dynamics as a behavioral biometric for authentication. In *ACM ASIACCS '11*.

[23] A. Karole, N. Saxena, and N. Christin. A comparative usability evaluation of traditional password managers. In *ICISC*, 2010.

[24] LastPass. Password manager. https://www.lastpass.com/.

[25] B. McGillion, T. Dettenborn, T. Nyman, and N. Asokan. Open-

tee–an open virtual trusted execution environment. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*.

[26] R. H. Morris and K. Thompson. Password security - a case history. *Commun. ACM*.

[27] J. Nielsen. 10 usability heuristics for user interface design. *Nielsen Norman Group*, 1(1), 1995.

[28] L. O'Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 2003.

[29] D. Recordon and D. Reed. Openid 2.0: A platform for user-centric identity management. In *ACM DIM '06 Workshop*.

[30] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell. Stronger password authentication using browser extensions. In *USENIX Security Symposium*, pages 17–32. Baltimore, MD, USA, 2005.

[31] J. Rubin and D. Chisnell. *Handbook of usability testing: howto plan, design, and conduct effective tests*. 2008.

[32] S. Ruj, M. Stojmenovic, and A. Nayak. Privacy preserving access control with authentication for securing data in clouds. In *IEEE/ACM CCGRID '12*.

[33] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit authentication through learning user behavior. In *ICISC'10*.

[34] F. Stajano. Pico: No more passwords! In *Security Protocols XIX*.

[35] G. Venkatadri, O. Goga, C. Zhong, B. Viswanath, K. P. Gummadi, and N. Sastry. Strengthening weak identities through inter-domain trust transfer. In *Proceedings of the 25th International Conference on World Wide Web*.

[36] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: Empirical results. *IEEE Security & privacy*, 2(5):25–31, 2004.

[37] R. Zhao, C. Yue, and K. Sun. Vulnerability and risk analysis of two commercial browser and cloud based password managers. 2013.

[38] N. Zheng, K. Bai, H. Huang, and H. Wang. You are how you touch: User verification on smartphones via tapping behaviors. In *IEEE ICNP '14*.