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

Kostantinos Papadamou
Cyprus University of Technology

Savvas Zannettou
Cyprus University of Technology

Giuseppe Bianchi
University Rome Tor Vergata

Alberto Caponi
University of Rome Tor Vergata

Annamaria Recupero
University of Rome Sapienza

Steven Gevers
Verizon Enterprise Solutions

George Gugulea
certSIGN

Sorin Teican
certSIGN

Bogdan Chifor
certSIGN

Christos Xenakis
University of Pireaus

Michael Sirivianos
Cyprus University of Technology

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: (a) the seamless integration between usable/strong device-centric authentication methods and federated login solutions; (b) 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 (c) 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 on the web heavily 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 ($\sim$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 easily verify the guessed password by performing a dictionary-based brute-force attack. Over the years, the scientific community repeatedly pinpointed the flaws of the password paradigm [15, 43, 9, 29].

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 password for each service. As a consequence, users resort in re-using the same password for each service they maintain [34]. Second, there is lack of support for Attribute Based Access Control (ABAC), which facilitates account-less access 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: (i) federated authentication solutions (i.e., OpenID Connect [30]) 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); (ii) strong and usable password-less authentication mechanisms, such as FIDO UAF [11]); and (iii) cryptographic credential stacks, such as Idemix [17] and U-Prove [44] that facilitate Privacy-preserving Attribute-based Access Control (PABAC). 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

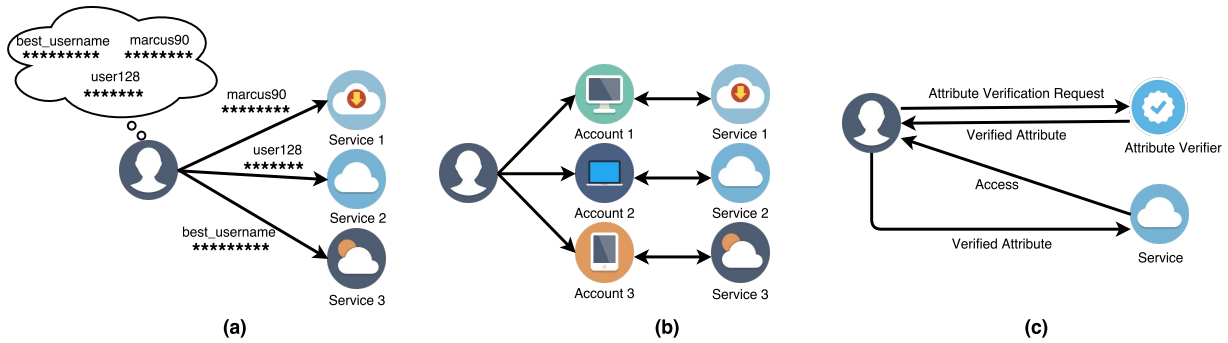

Figure 1: Caveats of the prevalent web authentication paradigm (a) password overload; (b) identity fragmentation and (c) lack of support for Attribute-based Access Control

specialized components within their infrastructure.

Other studies [46, 38, 6] propose the use of password managers, which enables 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 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 end-users 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 [11] for strong authentication mechanisms, and from the OpenID Foundation [30] for federated authentication. This integration enables a federated authentication solution where end-users 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) [33], which is equal to a secure SIM authenticator. Therefore, promoting the device to the main authentication gateway not only ease the user from the burden of remembering multiple complex passwords, but also facilitates technically complex but needed authentication modalities that make our architecture fully aligned with the latest NIST standards for authentication.

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 and 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 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). 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, whereas BAA can ensure that unauthorized access to services by illegitimate holders of the device is prevented.

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. This architecture provides a rich set of features to the end-user through the IDC. Specifically, an end-

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, behavioral and MC authentication.

Organization. In Section 2, we define important terminology and the required background. In Section 3 we define the threat model and the requirements that enable the design of our architecture. Section 4 provides a description of the main components that comprise our proposed architecture. In Section 5 we report the design of our architecture while in Section 6 we describe our prototype implementation. In Section 7 we evaluate the performance of our prototype implementation. 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). 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 [39]. 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 the authentication of end-users. In the context of Privacy-Preserving Attribute-based Access Control, IdPs are responsible for issuing and verifying end-users' cryptographic credentials.

Identity Consolidator (IDC). This is a centralized trusted entity that manages all the access control needs of the user. The user is able to authenticate against the IDC, issue and verify cryptographic credentials, perform failure recovery (in case of lost or damaged device), lock/unlock its online accounts. For more information see Section 4.

Service Providers (SP). These are entities that are responsible only for the 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 aim to 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 are able to 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 changes in the SP's authentication stacks. 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 a 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 SP can obtain, with the user's explicit consent, identity attributes of the user 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 secure password-less solutions 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 service. When a user authenticates with her device using biometrics, he unlocks the stored cryptographic keys which are subsequently used for authentication against the service.

Privacy-Preserving Attribute-based Access Control (PABAC). Refers to the Authentication and Authorization procedure that is realized through the use of cryptographic credentials. We propose the use of state-of-the-art cryptographic credentials stacks that are integrated with Federated solutions. Specifically, we propose the use of Idemix [17] and U-Prove [44] cryptographic stacks, which are deployed on trusted entities (IdPs).

3 Threat Model and Requirements

In this section we define the threat model and the requirements for the proposed architecture. Both requirements

and threat model enable the design and definition of our architecture as described in Section 4 and Section 5.

3.1 Threat Model and Assumptions

The proposed architecture is vulnerable to various threats that must be identified. This will guide us to define the requirements and better design our architecture. 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. Initially, we admit that it can be stolen by an attacker. Assuming that the attacker is not able to perform software attacks, our architecture is able to effectively prevent this kind of threat by employing behavioral authentication and with a specialized account locking module that allows the lock access to her online accounts on a specific device. Specifically, the diverse behavioral authentication can prevent an attacker from being authenticated as the legitimate user.

Next, we also examine the case where the attacker is able to perform software attacks. If the behavior capturing protocols run in the Normal World (Rich OS), a skilled software attacker can intervene and modify the contents of the memory and also intercept or modify information from the device's sensors. If all the local measurements are immediately sent to the BAA then we can prevent such 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 keys and performing secure cryptographic operations. For example, the activation of behavioral authentication with a verifier can be triggered by TEE-enabled secure biometrics (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.

Furthermore, it is critical to take into account the case where the attacker, who has stolen the device, is able to perform hardware and software attacks. Behavioral authentication, as described above, protects the user in this scenario and the attacker can bypass the trusted authentication activation and present himself as the legitimate user only if the device is recently stolen.

Service and Identity Providers. Like every online service, the SPs in our architecture face various threats. First we have to prevent access tokens and server responses from disclosure during an authentication. Our solution guarantees that the authentication tokens are never exposed to unauthorized parties and all server responses are sent over TLS. Another threat that we identify is cross site request forgery (CSRF) and we defend against this threat by performing header checks in order to verify the origin of the source and destination for every request and we also make use of CSRF tokens in the communication between a user and a SP.

Besides the aforementioned, there are other types of attacks in the communication between a user and a SP like man-in-the-middle attack where the attacker intercepts this communication for various malicious purposes. An example of this is the access token redirect and reuse where the attacker can obtain and reuse the token the access token generated by the IdP in order to obtain additional access to resources. Additionally, the attacker can reuse one-time tokens like an authorization code that has already been used and gain access to the intended resources. We defend against such threats by generating access tokens that are user and scope restricted.

Another type of attack that we have to defend against is the pharming attack where the attacker corrupts an infrastructure service causing the user to be redirected to a fake SP. This attack can cause the user to reveal sensitive information to the attacker. Using our architecture to provide a trusted identity to SPs such an attack can be easily avoided. Additionally, the user device will authenticate against an IdP, using signed and encrypted JSON Web Tokens (JWTs) with an appropriate key and cipher.

TLS attacks and Replay attacks are also threats for our architecture. We defend against TLS attacks by deploying gateSAFE [19], which ensures a higher grade of security. In the case of a replay attack an eavesdropper intercepts a successful authentication of a legitimate user with an IdP before redirection to the SP. We make use of transport layer encryption for all the communications between the user device and the IdPs/SPs to prevent this kind of attack.

Furthermore, an IdP being compromised or acting maliciously is not considered since this is a general problem of federated architectures. If an IdP is compromised it only affects the authentication security on the SPs that relies on that IdP.

User Privacy. User privacy is of vital importance in our architecture. SPs may require authentication or accountability of users' actions when they need to prove their identity, or at least possession of a certificate. These SPs can identify a user through a combination of context from a series of transactions. Moreover, even if standard anonymization practices are performed by the user, if the issuer and the verifier are colluding, the user can be identified. Our PABAC approach enable us to preserve users'

privacy by employing advanced unlinkable and untraceable cryptographic credentials.

Additionally, the user is able to provide her consent when revealing identity attributes to SPs. This is achieved using the consent management module within the IDC. The IDC also includes the profile management module which provides to the users useful privacy risk indicators for each one of their attributes. These indicators define the risk of involuntary de-anonymization as well as the possibility of an attribute inference.

3.2 Requirements

In order to provide a complete solution and address all the aforementioned problems, our architecture should fulfill the following requirements:

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

R2: Ease of deployment. Incremental deployability is of vital importance 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. In order 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. Because of that, our architecture should offer additional authentication mechanisms to be triggered whenever SPs wish to further verify the identity of an end-user.

4 Architectural Overview

In this section we describe the main pillars of our architecture. This architecture consists of the following: (a) User Device; (b) Identity Consolidator; (c) Identity Provider; (d) Service Provider; and (e) 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 essential in our architecture as we aim to provide a 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 a FIDO UAF protocol stack we enable human-to-device authentication mainly 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 allows 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 stacks are also responsible for revealing issued credentials to IdPs during an authentication. Credentials stored in the CCS should not be exported even in the event that the device get compromised. This is achieved using a Secure OS along with hardware to set a TEE.

To enable continuous and second-factor authentication the mobile device includes a behavioral profile capture module, which is responsible to capture the behavior of the user taking advantage of the various sensors available in the 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 consid-

ered 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 user's request. The collected attributes are securely stored and maintained in an ID repository. The following modules comprises 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 OpenID Connect Provider for undertaking FIDO authentication. This module also allows the IDC to run federated login protocols for transferring identity attributes between different 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, SP, and IdP admins to register their entities with the IDC. Using this knowledge, the AMM can act as a BAA discovery service for the SPs that may require a second-factor authentication. Besides these, the AMM allows the users to manage their IDC account e.g., to set the preferred degree of privacy within the IDC or completely delete her account. Lastly, the AMM facilitates the integration of MC protocol within our architecture. To achieve this, IDC act as a relay for SPs that request MC authentication.

Credential Management Module (CMM). 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 with the IDC for failure recovery purposes. Credential management module 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 him 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 their consent for their various identity attributes.

Identity integration module (IIM). The main responsibility of this module is the standardization and normalization of the users' identity information collected from various IdPs. It 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. Several IdPs (e.g., universities) are involved when required by the end-users or the SPs. Each IdP has an identity repository that stores users' attributes. IdPs also run cryptographic credential stacks (i.e., Idemx and U-Prove) that facilitates 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 with the use of an Access Control Policy Reasoning tool. This tool is based on the defined AC policies to evaluate users' requests on resources based on the provided attributes. Besides this, it also recommends to administrators policy improvements derived from an underlying machine learning algorithm.

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-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 the OpenID Connect specification.

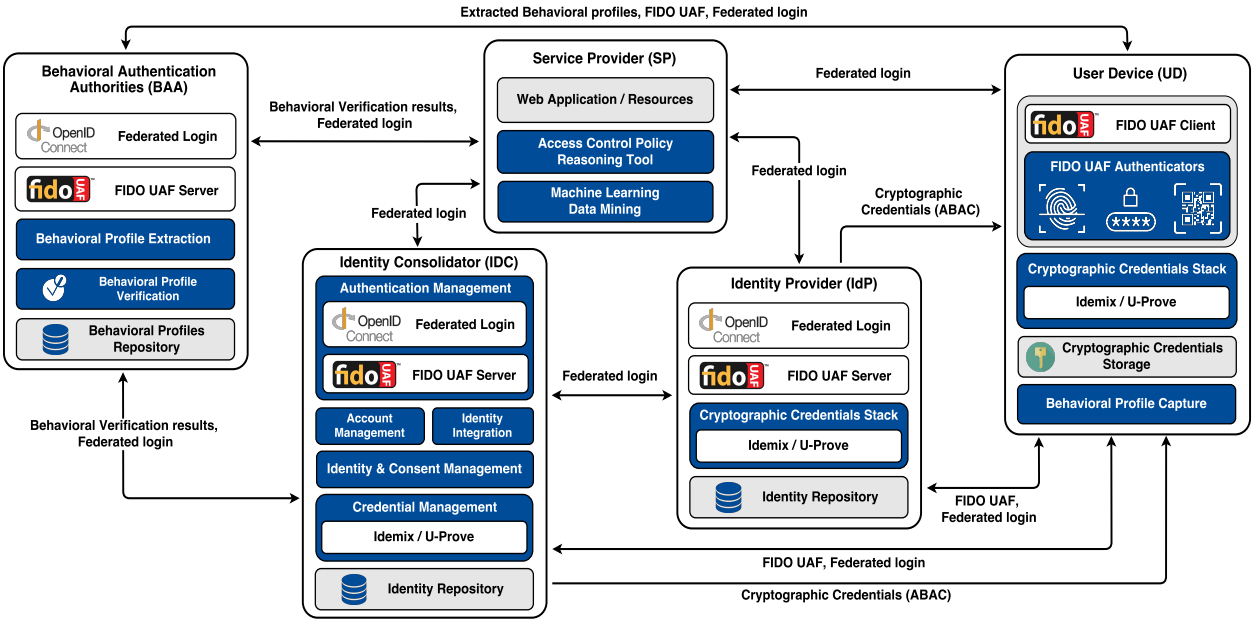

Figure 2: Privacy preserving architecture for device-centric and attribute-based authentication. The main architectural components with the modules that comprises each component.

The idea of using entities able to authenticate users based on their behavior was first proposed by Chow et al. [24]. In this work, BAAs are also separate entities able to perform behavioral authentication. However, since behavior is not privacy-preserving we offer behavioral authentication as a second-factor while the first-factor can be privacy-preserving using PABAC.

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

Within our architecture we enable PABAC by integrating the Idemix and U-Prove cryptographic credential stacks within the OIDC Provider on the IdPs. Users can request the issuance of cryptographic credentials by these IdPs or the IDC. This solution has various advantages which are: (a) SPs are not required to deploy any cryptographic credential stacks to support PABAC. Instead, they delegate the verification of PABAC credentials to IdPs; and (b) 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 cleanly 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 [4] diverse authentication framework for the end-users. Specifically, our federated architecture offers various authentication modalities. Depending on which is used the granted Authenticator Assurance Level (AAL) is determined. For example, the highest degree of assurance (AAL3) requires a hard 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 [1]) because of advances in TEE and TPMs.

Moreover, backup password along with behavioral authentication grants to the user the lower degree of assurance, which is AAL1, while FIDO UAF authentication alone provides AAL2.

5.2 FIDO-enhanced Federated Authentication

OpenID Connect [30] is a simple federated identity layer on top of the OAuth 2.0 protocol [5], which facilitates federated authentication. Thus, OIDC specification en-

ables 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 [11], which is a password-less solution that enables IdPs to authenticate end-users using strong authenticators, such as biometrics for user-to-device authentication (e.g., fingerprint) 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 OIDC Provider that acts as Idemix/U-Prove verifier, thus allowing IdPs to issue and verify cryptographic credentials.

Federated PABAC offers two concepts of anonymity, namely untraceability and unlinkability. In addition, users' privacy is preserved since they are able to authenticate to SPs by revealing only the required attributes without revealing their whole 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 gateway/proxy to SPs for discovering and contacting MC IdPs (Mobile Network Operators-MNOs) on behalf of the SP. In this way we make use of the MNOs as any other IdP within our architecture using OIDC. The IDC acts as a MC SP to retrieve the required attributes. After it retrieves the MC attributes of the user, the IDC acts as a vanilla OIDC provider that proves those attributes to another SP that is not registered in the MC ecosystem.

An advantage of this solution is that 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 Mobile Connect 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 easily be 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, which is the human-to-device biometric factor.

Using MC and BAA the user has to first login to the IDC using her secure backup password, which is used only in case of 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 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). 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 behavioural 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 access, the device sends behavioral records to the BAA, while all the records prior to the new device logging in are not considered for the authentication. The IDC acts as a SP while the BAA acts as an IdP authenticating the user based on her behavior. The IDC keeps the user under tentative access until she successfully proves to the BAA that she behaves as she always does. Once the BAA has collected sufficient records to give a ver-

dict 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 with 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 increase formally your NIST authenticator assurance level but it is just an extra assurance.

If the user does not desire to use any backup password, she is also able to recover from failure with physical identity verification. In this case the user is requested to scan her eID or ePassport using her mobile device. Taking advantage of 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 with full access to the IDC.

5.6 Multi-device Support

For usability purposes we identified the need to support multiple devices. Thus, we have 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 a 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.

Furthermore, we acknowledge that 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 produce the confidence probability of whether a SP can infer the value of an attribute that the user has not explicitly 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 a SP since ABAC prevents the SP from uniquely identifying the user. The calculation is made based on the attribute or combination of attributes that the user is about to share with a SP.

5.8 Deployability and Adoption

Federated architectures have many significant benefits for its adopters. First, user experience is enhanced since an end-user has to consolidate and proof his identity once at the IDC and then it can be reused to access multiple SPs. Second, there 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. Thus, SPs can focus on their mission rather than the business of identity management.

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 [2] for remote gambling in UK.

6 Implementation

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

FIDO UAF/OIDC: To exploit the OpenID Connect Provider features, we make use of the OpenAM software [1]. In our deployment, within the OpenID Provider

[1] https://forgerock.org/openam/

software, we have implemented 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, the custom authentication module communicates with the FIDO UAF Server using a REST interface. The FIDO UAF Server is responsible for performing the authentication of the user by communicating with the FIDO UAF client that runs on users' devices.

PABAC/OIDC: PABAC is realized through the deployment of Idemix/U-Prove cryptographic credential stacks. To enable IdPs to act as cryptographic credentials issuers/verifiers we have implemented a custom authentication module within OpenAM. For this purpose we use the FIWARE REST API specification [28], which is able to utilize both underlying cryptographic protocol stacks used in our architecture.

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

Moreover, we also implemented a Mobile Connect proxy service. In this context we implemented a custom module within the IDC that allows the IDC to act as Mobile Connect proxy. This custom module invokes a GSMA Apigee API Exchange-enabled [40] discovery service on a trusted Mobile Connect provider. This API is mainly used as the federation mechanism for Mobile Connect.

Behavioral Authentication: An important contribution that we make in this work is the stand-alone BAA entities. In order the BAAs to offer what is described in this work, we had to implement both the BAA server and client side modules that will capture and verify the behavior of the user. The captured behavior is transferred to the BAA through an authenticated session established between the user device and the BAA. Authentication is undertaken by the developed FIDO/OIDC custom authentication module, which we also integrate within our BAA implementation.

Furthermore, each BAA is responsible to integrate its own back-end logic (e.g., machine learning classifiers) within our BAA implementation as well as offering the appropriate behavior capturing android modules. In the context of this work we developed a BAA entity for gait verification including its custom android modules that captures the gait of the end-user on her device.

User Device: We implemented an Android application that integrates all the required user device functionality. This application runs a FIDO client, the behavior capturing module, and it utilizes the TEE to store cryptographic credentials. In general, we increase maintainability by implementing each module as a separate library integrating them all together in one application.

7 Evaluation

In this section we assess our prototype implementations in terms of performance and User Experience (UX). First, we evaluate the performance of our custom authentication module implementation. Then we evaluate our BAA prototype implementation in terms of efficiency (accuracy) and in the end we report the results of our UX evaluation, which is a work in progress.

7.1 FIDO UAF

As described in Section 6, we implemented a custom authentication module by deploying a FIDO server to the IdP's software stack to enable IdPs authenticate end-users using the FIDO UAF protocol. Here, we evaluate this FIDO server in terms of performance in order to identify how it performs under high registration/authentication demands. In the evaluation we do not count any user induced delays.

Fig. 3 presents the results of a FIDO registration process simulation. As it can be observed, our FIDO server implementation scales along with the number of users, with the server's average response time not being drastically impacted when the number of simultaneous requests is below 2400. The simulation was performed by porting our Android FIDO UAF client implementation on a desktop and by simulating the parallel FIDO registration processes using different threads. For the simulation we employed the FIDO registration process, but comparable results were obtained also for the FIDO authentication process, with these two processes employing similar cryptographic operations. Regarding the simulation results, we measured the average response time for each registration request (HTTPs request/response), this being the time for all FIDO messages to be exchanged between the client and the FIDO server. The total overhead of the FIDO client is not relevant in our context, because the clients can use mobile devices with different computing capabilities. The experiments were performed by using the FIDO server which runs in a docker container on the IdP and the client requests are executed using an Internet connection.

7.2 PABAC

For this evaluation we have also implemented a custom PABAC authentication module based on the U-Prove protocol. We have deployed the implemented module to the IdP's software stack in order to enable the IdP to act as a U-Prove credential verifier able to authenticate

users. The purpose of this evaluation is to identify how our PABAC-enabled IdP implementation performs under high authentication demands. In our measurements we do not consider the cost for issuing U-Prove cryptographic credentials.

Fig. 4 shows the results of the performance testing that we have performed by simulating the U-Prove authentication process. As it can be observed, our U-Prove server (verifier) implementation scales along with the number of requests, with the server's average response time not being drastically impacted when increasing the number of simultaneous requests except when the number of simultaneous requests exceeds 2500. Here, we measure the average response time for each U-Prove authentication request (HTTPs request/response), this being the time required for all messages to be exchanged between the U-Prove server and the client.

As with FIDO, the simulation was performed by porting our Android U-Prove client implementation on a desktop and by simulating the parallel U-Prove authentication processes using different threads. In addition, we use a U-Prove server which runs in a docker container on the IdP and the client requests are executed using an Internet connection. For a fair evaluation, we have hardcoded the U-Prove credential required for authentication in the code of the client and in the simulation we do not count the time required for the issuance of this credential. Also the total overhead of the U-Prove client is not relevant in our context, because the clients can use mobile devices with different computing capabilities.

7.3 Behavioral Authentication Authorities

In our architecture, we rely on BAAs for both continuous and on-demand second-factor authentication. To assess the feasibility, accuracy, and effectiveness of such design choice, we evaluate the prototype implementation of a proof-of-concept gait-based solution that we implemented as part of a BAA entity.

For the evaluation we have recruited 110 users (89 males, 15 females and the rest unknown) who voluntarily downloaded from Google Play store the gait-based BAA client that we have developed as an android application, which runs in the background of the device and continuously captures gait samples while the user is moving. The age of the participants varies from 20 to 60 years with a mean age of 30 years and a standard deviation of 6,8. We were collecting data for 2 weeks which yielded in 60k samples of 1-minute length data captured as walking periods. All the samples were stored and processed on our BAA server. For performance reasons we choose to use a Random Forests classifier [16] for our data classification process, which is an ensemble learning method for classification and regression with decision trees where the trees are constructed using bootstrap aggregation.

Impact of the walking portion included in samples on gait-based authentication. From the collected data, we first tried to determine how important is the length of a gait sample and identify the ideal length that will be the threshold for accepting a gait sample as a valid sample.

For any user to be a part of this experiment, she needed to have at least 25 samples that includes 100% walking and this reduced the number of selected users from the total 110 to 29. From each selected user, we then randomly selected 25 samples to be used for training our classifier and 5 for testing. We repeated the experiment by increasing the minimum required portion of walking in steps of 10%. From the results, we came to the conclusion that using samples with 70% or higher portion of walking we can successfully authenticate (uniquely identify) users with at least 80% accuracy as can be seen in Figure 5.

Minimum samples required for successful authentication. We also performed experiments in order to identify the minimum number of samples required to authenticate a user. A 70% walking percentage was used instead of 100% which resulted in 56 users to be selected for the experiment. We started by increasing gait sam-

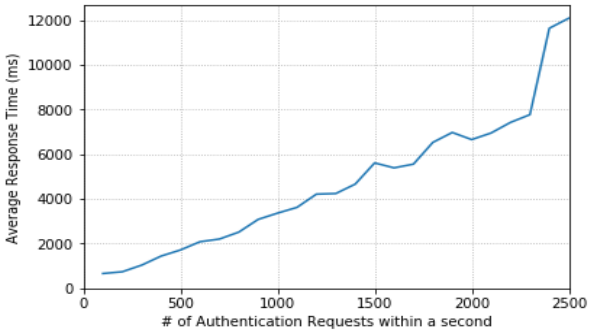

Figure 3: Average response time for a FIDO authentication/registration request as the number of requests increases.

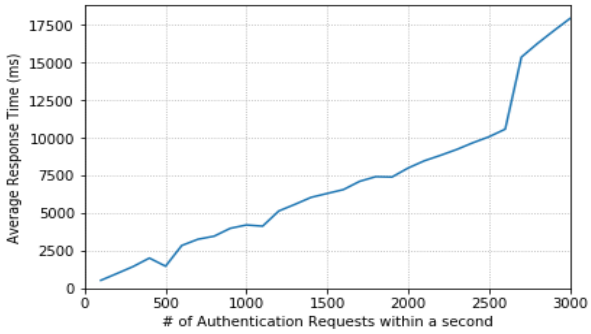

Figure 4: Average response time for a U-Prove authentication request as the number of requests increases.

Figure 5: Portion of walking included in each gait sample. As the portion increases results in improved authentication accuracy.

ples from 1 to 25 to train our classifier that is responsible for predicting whether a gait sample matches the gait of a specific user. The samples were selected in consecutive time order but with a random starting point. The experiment was repeated 20 times and a mean score of the 20 experiments was calculated. From the results we came to the conclusion that requiring at least 20 gait samples we authenticate users with a balanced accuracy of 75% as depicted in Figure 6.

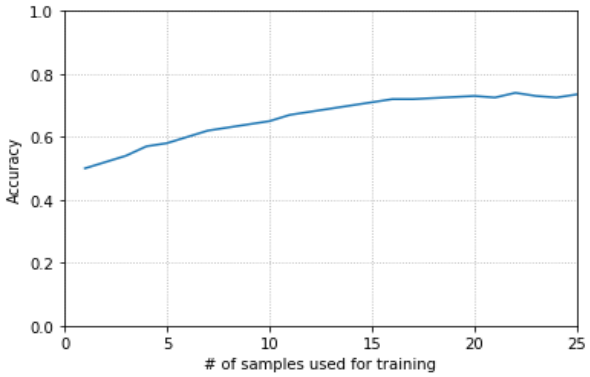

Figure 6: Number of samples required for training. As the number of samples increases results in better accuracy.

7.4 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. 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" [10]. 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: (a) preliminary assessment by experts, who analyze the level of compliance with the main usability heuristics [42] and identify usability issues to fix; (b) 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 [47]; and (c) 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 were recruited from the Electrical Engineering and Computer Engineering and Informatics Department of Cyprus University of Technology. The results of the survey provide positive feedback supporting the design, and suggestions to improve the UX. Regarding the usefulness of the proposed solution and its impact on the UX, most of the respondents (33 out of 41) agree that the solution provides a reliable and secure authentication mechanism, and makes access to online services easier and quicker.

Heuristics	**Positive (%)**	**Negative (%)**
Easiness	90	9
System Stat. Visib.	88	11
Language	92	8
Contr. & Error Prev.	93	7

Table 1: Usability heuristics and the percentage of positive/negative evaluation. Easiness: how easy is to perform the required tasks to accomplish a goal. Visibility of the system status: the system keeps users informed about what is going on through appropriate feedback. Language: the language used (e.g., words, messages, etc.) is clear and familiar. Control and error prevention: the system enables the user to easily diagnose and correct errors.

Considering the impact on the UX, most of the respondents (from the age of 31 to 39) 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 (from the age of 34 to 39) consider the language (i.e. terms, icons, messages) familiar and comprehensible, and the organization of the information clear. 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.

8 Related Work

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

8.1 Password alternatives

It is evident that nothing is more secure than a 128-bit ($\sim$20 characters) random password used for a specific service. However, if a password is secure enough then it is 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 be able to memorize multiple complex and long enough passwords for each service they use.

The past few years, the research community realized that the password paradigm is not an ideal solution that is able to cope with users' authentication needs on the Web; mainly because of usability and security concerns. Therefore, various works aim at either replacing the password paradigm or propose solutions that mitigate its caveats. Specifically, [9, 41, 55] identify and 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 [46], LastPass [38], and RoboForm [6] 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 [23, 56] 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 in case of failure recovery and not every time the user wants to authenticate with a service.

Additionally, passwords managers cannot protect the user against 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. Also, in the case of a breach attack the compromised service only contains a perfectly useless list of public-keys. Another advantage of our approach is that each private-key can be as long and random as needed to stay secure.

On top of the aforementioned, Karole et al. [37] highlight users' concerns with regard to the use of password managers. That is they feel more confident in storing their passwords locally instead of storing them in cloud-based password managers.

A solution that aims at addressing the usability issues is the one by Chatterjee et al. [22] who proposes TypTop, a typo-tolerant password checking system tailored to the user behavior. It allows users to submit and use a password that contains a small number of typographical errors. However, the usability gains from this approach are negligible.

Furthermore, there are also studies that propose alternatives to the password paradigm. Stajano [52] 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. Pico is resistant to theft and loss using Picosiblings, which are smaller tokens that communicate with Pico and are used as an unlock mechanism. In contrast to Pico, our architecture does not introduce any extra hardware. Instead, we leverage users' mobile device to store the necessary cryprographic keys for authentication purposes.

Trusona [8] is a product that uses an architecture relavant to the one we propose. It offers device-centric password-less and multi-factor authentication through the use of a mobile application. A user can register by scanning one of her identity documents to prove her identity. Their solution is suitable for various use-cases ranging from online authentication to wire transfers. To achieve this, Trusona offers up to four-factor authentication including biometrics (e.g., fingerprint), scanning of identity documents, hardware tokens, etc. With the IDC, we go beyond Trusona by offering sophisticated identity federation and proofing algorithms, as well as continuous behavioral authentication.

8.2 Behavioral Authentication

Behavioral Authentication can provide 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 [57, 26].

Google has also identified the need for replacing the

password with stronger and more usable authentication mechanisms. To this end, Google provides the Trust API [31], 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 [36, 27, 51, 35]. However, the classifier's location in most of these approaches is not specified and they do not consider battery, computational and space limitations. Yet, 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. We also offer enhanced protection against attackers that manage to compromise the device because the behavioral record of a user is stored on the BAA side rather than locally. In addition we perform continuous behaviour authentication that allow us to have account-wide and device-wide lockdown when the device is not held by its legitimate user.

Chow et al. [24] 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 [24] by offering BAAs as part of an entire DCA and identity consolidation solution. In addition, BAAs perform continuous authentication to detect when a device is compromised.

Enterprises like [50, 12, 14, 49] offer continuous behavioral authentication software as a service. They use real-time behavioral and statistical analysis tools to detect automated and human 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 a SP deems necessary, spanning from a simple boolean flag to statistical scores.

8.3 Identity Federation and Management

Identity federation and management is essential for a DCA architecture for various reasons. First, it solves the identity fragmentation problem. Second, it is required for efficient failure recovery; and third it enables users to have complete control over their identity information and privacy.

The past two decades a lot of identity federation and management solutions have emerged. One of them is the WSO2 Identity Server [54], which is an open source technology that when integrated within a SP's infrastructure can offer singe sign-on (SSO), and identity federation and management. WSO2 acts as a standard IdP authenticating users using multi-factor authentication and it can also delegate the authentication to other IdPs.

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. They can also request behavioral authentication, and MC authentication in case they want to achieve a high assurance authentication level.

OpenID 2.0 [45] is a user-centric identity management platform in which each account has Identifiers (URI) at one or multiple IdPs. It enables an end-user to prove the possession of an identifier. These identifiers are send to SPs and then mapped to the appropriate IdP. 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. In this model, users must fully trust both the IdP and the SPs.

Other identity management approaches like Liberty [3] and SAML [18], offer federated user identities in a more privacy-preserving way. IdPs use pseudonyms or aliases to reference users to the SPs' namespaces and these pseudonyms are different in each SP. This entails that 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 with cryptographic credential stacks to employ pseudonyms so that user anonymity is maintained.

Venkatadri et al. [53] 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. In our architecture we deploy more sophisticated algorithms for assessing the trustworthiness of a user's identity with high confidence.

A more akin to our architecture solution is the Secure Identity Platform (SIP) by Civic [7]. 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 and at the same time with PABAC the privacy of

the end-user is preserved. However, inspired from SIP's decentralized architecture we consider as future work the possible decentralization of the IDC component of our architecture taking advantage of the blockchain technology.

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 [32] uses the policies to create the applicant keys and uses the attributes to describe the encrypted data. Ciphertext-policy ABE [13] uses a tree form access policy, where attributes are the leaves of the tree.

Ruj et al. [48] propose a privacy-preserving access control scheme in the clouds, in which the attributes of each user belong to multiple key distribution centers (KDC) [25]. The user's identity information that is stored in the cloud is not known by the cloud. The cloud acts as the verifier for the users' credentials. However, user privacy is not protected in the cloud and the cloud knows the access policy tables.

Chase [20] 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 [21] 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 [17] and U-Prove [44]) to let users prove their identity attributes to SPs using cryptographic credentials that are securely stored on their device. In addition, 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 a substitute for SPs that want 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.

10 Acknowledgments

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] Fips 140-2, security requirements for cryptographic modules. https://csrc.nist.gov/publications/detail/fips/140/2/final.

[2] Gambling Act 2005 legislation. https://www.legislation.gov.uk/ukpga/2005/19/contents.

[3] Liberty alliance. http://www.projectliberty.org.

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

[5] OAuth 2.0. https://oauth.net/2/.

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

[7] Secure identity platform by civic. https://www.civic.com/products/secure-identity-platform.

[8] Trusona. https://www.trusona.com.

[9] ADAMS, A., AND SASSE, M. A. Users are not the enemy. *Communications of the ACM*.

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

[11] ALLIANCE, F. Fido alliance specifications overview. https://fidoalliance.org/specifications/overview.

[12] BEHAVIOSEC. Continuous authentication. https://www.behaviosec.com.

[13] BETHENCOURT, J., SAHAI, A., AND WATERS, B. Ciphertext-policy attribute-based encryption. In *IEEE SP '17*.

[14] BIOCATCH. Behavioral biometrics fraud prevention and detection. http://www.biocatch.com.

[15] BONNEAU, J., HERLEY, C., OORSCHOT, P. C. V., AND STAJANO, F. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *IEEE SP '12*.

[16] BREIMAN, L. Random forests. *Machine Learning* (2001).

[17] CAMENISCH, J., AND VAN HERREWEGHEN, E. Design and implementation of the idemix anonymous credential system. In *ACM CCS '02*.

[18] CANTOR, S., KEMP, J., PHILPOTT, R., AND MALER, E. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. Tech. rep.

[19] CERTSIGN. gatesafe. https://www.certsign.ro/certsign_en/produse/siguranta-organizatiei/gatesafe.

[20] CHASE, M. Multi-authority attribute based encryption.

[21] CHASE, M., AND CHOW, S. S. M. Improving privacy and security in multi-authority attribute-based encryption. In *ACM CCS '09*.

[22] CHATTERJEE, R., WOODAGE, J., PNUELI, Y., CHOWDHURY, A., AND RISTENPART, T. The typtop system: Personalized typo-tolerant password checking. In *CCS* (2017).

[23] CHIASSON, S., VAN OORSCHOT, P. C., AND BIDDLE, R. A usability study and critique of two password managers. In *USENIX Security Symposium* (2006).

[24] CHOW, R., JAKOBSSON, M., MASUOKA, R., MOLINA, J., NIU, Y., SHI, E., AND SONG, Z. Authentication in the clouds: A framework and its application to mobile users. In *ACM CCS '10 Workshop*.

[25] D'ARCO, P., AND STINSON, D. R. On unconditionally secure robust distributed key distribution centers. In *ASIACRYPT* (2002).

[26] DE LUCA, A., HANG, A., BRUDY, F., LINDNER, C., AND HUSSMANN, H. Touch me once and i know it's you!: Implicit authentication based on touch screen patterns. In *SIGCHI '12*.

[27] FENG, T., LIU, Z., KWON, K. A., SHI, W., CARBUNAR, B., JIANG, Y., AND NGUYEN, N. Continuous mobile authentication using touchscreen gestures. In *2012 IEEE HST*.

[28] FIWARE RESTful API Specification. https://goo.gl/dkG5R8, 2013.

[29] FLORENCIO, D., AND HERLEY, C. A large-scale study of web password habits. In *WWW'07*.

[30] FOUNDATION, O. Openid connect specification. http://openid.net/connect.

[31] GOOGLE. Android's trust api: a short history, and why it's a game changer. https://thisdata.com/blog/androids-trust-api-a-short-history-and-why-its-a-game-changer.

[32] GOYAL, V., PANDEY, O., SAHAI, A., AND WATERS, B. Attribute-based encryption for fine-grained access control of encrypted data. *IACR Cryptology ePrint Archive* (2006).

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

[34] IVES, B., WALSH, K. R., AND SCHNEIDER, H. The domino effect of password reuse. *Communications of the ACM 47*, 4 (2004), 75–78.

[35] JAKOBSSON, M., SHI, E., GOLLE, P., AND CHOW, R. Implicit authentication for mobile devices. In *USENIX HotSec'09*.

[36] JORGENSEN, Z., AND YU, T. On mouse dynamics as a behavioral biometric for authentication. In *ACM ASIACCS '11*.

[37] KAROLE, A., SAXENA, N., AND CHRISTIN, N. A comparative usability evaluation of traditional password managers. In *ICISC* (2010).

[38] LASTPASS. Password manager. https://www.lastpass.com/.

[39] MCGILLION, B., DETTENBORN, T., NYMAN, T., AND ASOKAN, N. Open-tee–an open virtual trusted execution environment. In *Trustcom/BigDataSE/ISPA, 2015 IEEE* (2015).

[40] MOBILE CONNECT, A. Gsma apigee api exchange. https://apigee.com/about/tags/api-exchange.

[41] MORRIS, R. H., AND THOMPSON, K. Password security - a case history. *Commun. ACM 22* (1979), 594–597.

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

[43] O'GORMAN, L. Comparing passwords, tokens, and biometrics for user authentication. *IEEE* (2003).

[44] PAQUIN, C., AND ZAVERUCHA, G. U-prove cryptographic specification v1.1 (revision 3). https://www.microsoft.com/en-us/research/publication/u-prove-cryptographic-specification-v1-1-revision-3/, December 2013.

[45] RECORDON, D., AND REED, D. Openid 2.0: A platform for user-centric identity management. In *ACM DIM '06 Workshop*.

[46] ROSS, B., JACKSON, C., MIYAKE, N., BONEH, D., AND MITCHELL, J. C. Stronger password authentication using browser extensions. In *USENIX Security Symposium* (2005), Baltimore, MD, USA, pp. 17–32.

[47] RUBIN, J., AND CHISNELL, D. *Handbook of usability testing: howto plan, design, and conduct effective tests*. John Wiley & Sons, 2008.

[48] RUJ, S., STOJMENOVIC, M., AND NAYAK, A. Privacy preserving access control with authentication for securing data in clouds. In *IEEE/ACM CCGRID '12*.

[49] SECUREAUTH. Behavioral biometrics. https://www.secureauth.com/products/secureauth-idp/behavioral-biometrics.

[50] SECURITY, N. Positively verifying users. https://nudatasecurity.com/.

[51] SHI, E., NIU, Y., JAKOBSSON, M., AND CHOW, R. Implicit authentication through learning user behavior. In *ICISC'10*.

[52] STAJANO, F. Pico: No more passwords! In *Security Protocols XIX*.

[53] VENKATADRI, G., GOGA, O., ZHONG, C., VISWANATH, B., GUMMADI, K. P., AND SASTRY, N. Strengthening weak identities through inter-domain trust transfer. In *Proceedings of the 25th International Conference on World Wide Web* (2016), WWW '16, International World Wide Web Conferences Steering Committee.

[54] WSO2. Identity and access management. http://wso2.com/identity-and-access-management.

[55] YAN, J., BLACKWELL, A., ANDERSON, R., AND GRANT, A. Password memorability and security: Empirical results. *IEEE Security & privacy 2*, 5 (2004), 25–31.

[56] ZHAO, R., YUE, C., AND SUN, K. Vulnerability and risk analysis of two commercial browser and cloud based password managers.

[57] ZHENG, N., BAI, K., HUANG, H., AND WANG, H. You are how you touch: User verification on smartphones via tapping behaviors. In *IEEE ICNP '14*.