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

Kostantinos Papadamou, Savvas Zannettou, Bogdan Chifor, Sorin Teican, George Gugulea, Alberto Caponi, Annamaria Recupero, Claudio Pisa, Giuseppe Bianchi, Steven Gevers, Christos Xenakis, and Michael Sirivianos

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.

I. Introduction

Authentication on 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 ($\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 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 [1]–[4].

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 online service (see Fig. 1(a)). As a consequence, they choose easy to remember passwords or resort in re-using the same password across multiple domains [5]. Second, a user's identity is fragmented across multiple services and there is not an easy way for them to prove account joint-ownership (see Fig. 1(b)). Last, there is lack of support for Attribute Based Access Control (ABAC), which facilitates account-less authentication through identity attributes (i.e., age or location); see Fig. 1(c). As a result, users are required to reveal multiple aspects of their identity even on services that may only need to verify their age.

Recent efforts aim at mitigating the aforementioned problems by proposing dedicated solutions. Specifically: 1) federated authentication solutions (i.e., OpenID Connect [6]) 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 [7]; and 3) cryptographic credential stacks that facilitate Privacy-preserving Attribute-based Access Control (PABAC) such as Idemix [8] and U-Prove [9]. 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 [10]–[12] propose the use of password managers, which enable users 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 that 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.

However, 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 [7] for strong authentication mechanisms, and from the OpenID Foundation for federated authentication [6]. This integration enables a federated authentication solution where users are able to authenticate using biometrics (e.g., fingerprint). 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

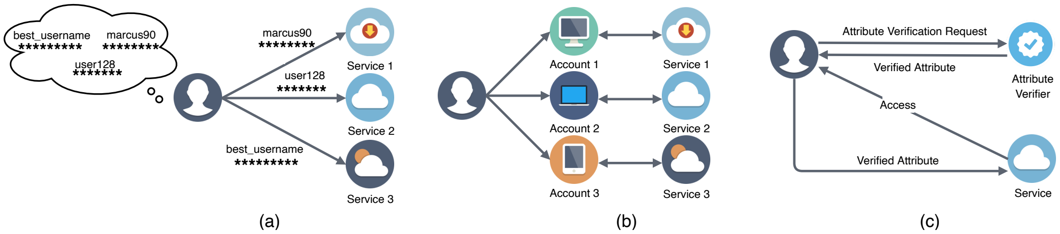

Fig. 1: Caveats of the prevalent Web authentication paradigm: (a) password overload; (b) identity fragmentation; and (c) lack of support for Attribute-based Access Control (ABAC).

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 (e.g., gait, keystroke, etc.). For increased assurance we employ Mobile Connect (MC) [13], 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 [14].

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 issues, we propose a reliable failure recovery framework by leveraging an innovative 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. Importantly, besides failure recovery, the IDC also offers real-world and online identity acquisition, identity and privacy management, and allows to prove combinations of fragmented identity aspects, thus solving the identity fragmentation problem.

Contributions. In summary, 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 under a privacy-preserving architecture.
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 propose an innovative architectural component, called Identity Consolidator, that solves the identity fragment problem and provides a rich set of features to the user. Specifically, a user can manage the spectrum of his online accounts and define options that will enhance his security, privacy and user experience on the Web.
5) We propose an innovative failure recovery framework, which is realized through the IDC, and behavioral and MC authentication.

II. Terminology

User Device. 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 (TEE) [15]. This enables the device to securely safeguard cryptographic credentials within its software stack.

Identity Providers (IdP). IdPs 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.

III. Threat Model and Requirements

In this section we define the threat model and the requirements that guide the design and definition of our architecture.

A. 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 access 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. User's privacy is of vital importance in our architecture. A malicious SP is in the position to infer a user's identity by combining identity attributes revealed in a series of distinct 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.

B. 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. 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 and 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 his identity attributes.

R4: Failure Recovery. All user access control needs should be anchored to his device, which enables authentication with various usable and cryptographically strong methods. The appropriate failure recovery mechanisms should be supported in case of device loss 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, 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 from their users for higher assurance. Hence, the proposed architecture should offer additional authentication mechanisms for SPs that wish to further verify the identity of a user.

IV. 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 the proposed architecture including its main components and the interfaces that interconnects them. All the communications between the components are built around the OIDC protocol by switching SP and IdP roles.

A. User Device (UD)

The mobile device of the user is central in our architecture as we aim to provide DCA. 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 (OIDC) with IdPs and SPs (aka, relying parties) for authorization and authentication purposes.

We also deploy cryptographic credential stacks (Idemix and U-Prove) on the device to enable PABAC. These stacks allow users to request the issuance of cryptographic credentials from the IDC and/or their IdPs and are responsible for revealing issued credentials to IdPs during an authentication. The issued credentials are stored in a secure fashion in the Cryptographic Credentials Storage (CCS) that is also part of the device. Using a Trusted Execution Environment (TEE) we ensure that credentials stored in the CCS cannot be exported even if the device has been compromised.

Last, to enable continuous and second-factor authentication, the software running on the mobile device includes a module that is responsible for capturing the behavior of the user taking advantage of the various sensors available on the device.

B. Identity Consolidator (IDC)

The IDC is an integral component in our architecture that fullfils the needs of requirement R3. It is a centralized fully trusted entity that can be considered as a special instance of an IdP, which offers identity federation, 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. We describe below all the modules that comprise the IDC.

Authentication Management Module (AuthMM). It encapsulates a FIDO-enhanced federated login protocol, which allows the IDC to act as an OIDC IdP for undertaking

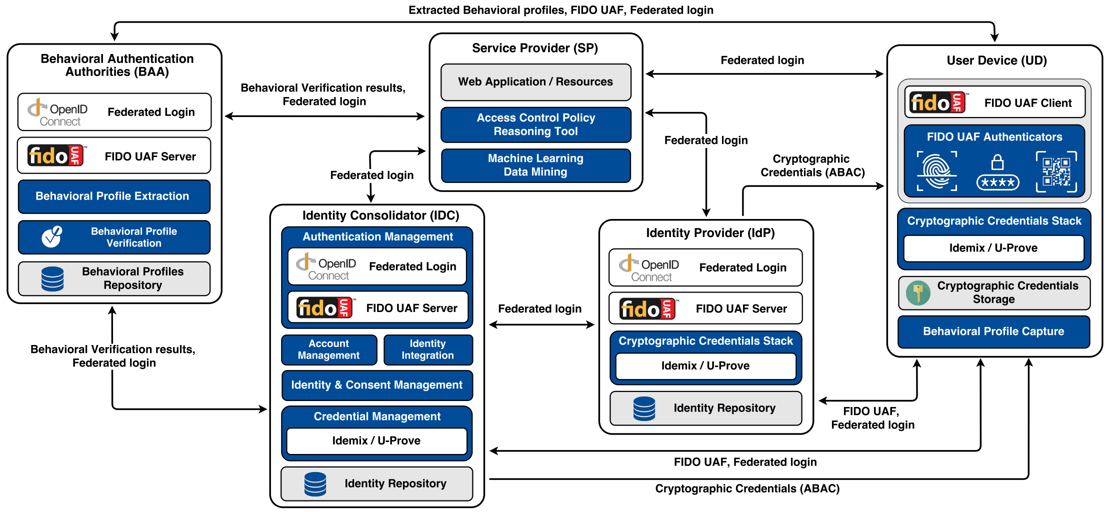
Extracted Behavioral profiles, FIDO UAF, Federated login
Behavioral Authentication Authorities (BAA)
OpenID Connect
Federated Login
fido UAF
FIDO UAF Server
Behavioral Profile Extraction
Behavioral Profile Verification
Behavioral Profiles Repository
Behavioral Verification results, Federated login
Service Provider (SP)
Web Application / Resources
Access Control Policy Reasoning Tool
Machine Learning Data Mining
Federated login
User Device (UD)
FIDO UAF Client
FIDO UAF Authenticators
Cryptographic Credentials Stack
Idemix / U-Prove
Cryptographic Credentials Storage
Behavioral Profile Capture
Federated login
Cryptographic Credentials (ABAC)
Identity Consolidator (IDC)
Authentication Management
OpenID Connect
Federated Login
FIDO UAF Server
Account Management
Identity Integration
Identity & Consent Management
Credential Management
Idemix / U-Prove
Identity Repository
Federated login
Identity Provider (IdP)
OpenID Connect
Federated Login
FIDO UAF Server
Cryptographic Credentials Stack
Idemix / U-Prove
Identity Repository
Behavioral Verification results, Federated login
FIDO UAF, Federated login
FIDO UAF, Federated login
Cryptographic Credentials (ABAC)

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

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 his device.

Account Management Module (AMM). 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 acts as a BAA discovery service for SPs that may require a second-factor authentication. In addition, the AMM enables the user to: 1) manage his IDC account, for example to set his preferred degree of privacy within the IDC (e.g., only certain attribute are stored on the IDC) or completely delete his account; 2) manage the status of his accounts in various SPs and IdPs; and 3) protect his accounts by locking access to them in case of device loss. The IDC can also act on behalf of the user and lock his online accounts when it detects a high risk of account compromise. Last, the AMM facilitates the integration of MC within our architecture. To achieve this, IDC act as a relay for SPs that request MC authentication (see Subsection V-D).

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 CMM also enables cryptographic credentials management, and allows users to backup their issued credentials at the IDC and restore them anytime on another device in case of device loss or failure.

Identity Management Module (IMM). IMM consists of the profile and the consent management modules that empower users to manage their identity information. The first module 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. The latter 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 this information via physical means (e.g., using Near Field Communication (NFC) 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.

C. *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 credentials from the stored identity attributes.

D. *Service Providers (SP)*

SPs require minimal modifications. Namely, they only have to run an OIDC client 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 (AC) policies. These policies can be managed by the SP administrator using an Access Control Policy Reasoning tool, which is also responsible to evaluate users' requests on resources based on the defined AC policies.

E. *Behavioral Authentication Authorities (BAA)*

BAAs are separate entities that provide both on-demand and continuous behavioral authentication as part of an entire DCA

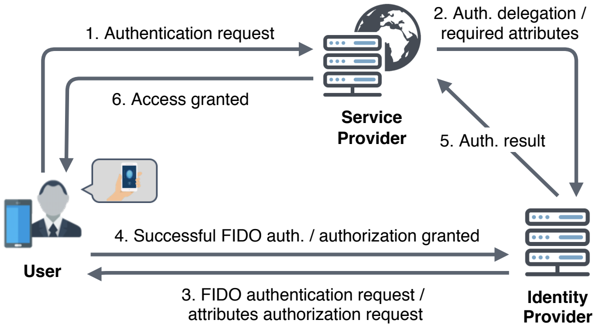
1. Authentication request
2. Auth. delegation / required attributes
6. Access granted
Service Provider
5. Auth. result
User
4. Successful FIDO auth. / authorization granted
3. FIDO authentication request / attributes authorization request
Identity Provider

Fig. 3: FIDO-enhanced federated authentication process.

solution. 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 his usual habits. The behavioral authentication outcome is released to the aforementioned entities using OIDC.

F. Privacy-Preserving Attribute-based Access Control (PABAC)

We enable PABAC by integrating the Idemix [8] and U-Prove [9] 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.

V. Design

In this section, we provide adequate information regarding the design of our 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 enables us to clearly separate the concerns of SPs and IdPs during an authentication, thus addressing requirements R1 and R2.

A. Diverse Authentication Framework

We propose a NIST-compliant [14] diverse authentication framework, thus addressing requirement R6. 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, a backup password along with behavioral authentication provides the lower degree of assurance (AAL1), while FIDO authentication alone provides AAL2. 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). We assume that in the future FIDO and MC will be as secure as a hardware cryptographic token (FIPS 140-2[1]) because of advances in the TEE.

B. FIDO-enhanced Federated Authentication

OpenID Connect (OIDC) is a simple federated identity layer on top of the OAuth 2.0 protocol [16], which facilitates federated authentication. OIDC 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 UAF specification 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 (e.g., RSA) for device-to-service authentication.

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. Fig. 3 depicts the proposed FIDO-enhanced federated authentication process. Initially, when the user tries to authenticate with an SP (step 1), the SP delegates the authentication to an IdP along with a list of identity attributes that the SP requires (step 2). Then, the IdP requests from the user to authenticate using FIDO on his mobile device (e.g., fingerprint). The IdP also requires from the user authorization to reveal to the SP the requested identity attributes (step 3). As soon as FIDO authentication is successful and the user has authorized the revelation of the requested attributes (step 4), the IdP informs the SP about the result of the authentication while also providing the requested attribute values (step 5). At the end, the SP grants to the user access to its resources (step 6).

C. Federated Privacy-preserving Attribute-based Authentication

Our architecture was carefully designed to provide a PABAC solution on top of OIDC, while also addressing requirement R5. More precisely, we propose a custom authentication module within the OIDC Provider that acts as an Idemix/U-Prove verifier allowing IdPs to issue and verify cryptographic credentials. In fact, we modify the OIDC Provider so that it uses one-time pseudonyms instead of persistent unique identifiers. In this way, we enable SPs that are not aware of any cryptographic credentials stacks to allow end-users use cryptographic credentials and get access to their resources.

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 only revealing the required attributes without revealing their complete identities.

[1] https://csrc.nist.gov/publications/detail/fips/140/2/final

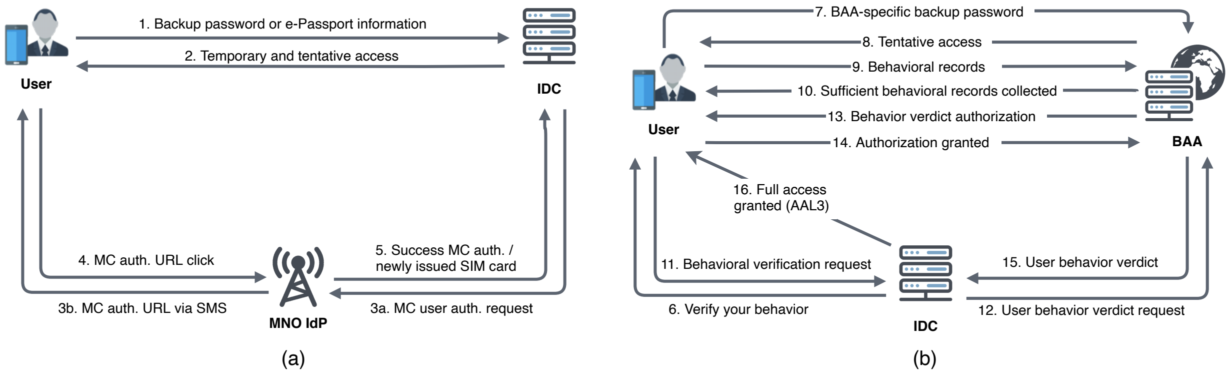
1. Backup password or e-Passport information
2. Temporary and tentative access
User
IDC
4. MC auth. URL click
5. Success MC auth. / newly issued SIM card
3b. MC auth. URL via SMS
MNO IdP
3a. MC user auth. request
(a)
7. BAA-specific backup password
8. Tentative access
9. Behavioral records
10. Sufficient behavioral records collected
13. Behavior verdict authorization
14. Authorization granted
User
BAA
16. Full access granted (AAL3)
11. Behavioral verification request
15. User behavior verdict
6. Verify your behavior
IDC
12. User behavior verdict request
(b)

Fig. 4: Failure recovery framework: (a) shows the first part of the failure recovery that involves MC authentication and (b) shows the second part that involves the verification of the user's behavior through a BAA.

Idemix provides both untraceability and unlinkability, while U-Prove provides only untraceability [17]. However, the richer feature set of Idemix comes at a performance cost. For this reason, we offer both Idemix and U-prove and we allow end-users to choose which one they prefer based on their needs.

D. 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 (MNOs) on behalf of the SP. Using OIDC, we see MNOs as any other IdP within our architecture. The IDC acts as an MC SP to retrieve the required attributes, and then acts as an IdP 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.

E. 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. First, in case the device is stolen, the thief has direct access to the secret. We address this problem via FIDO on devices. However, the most crucial problem involves recovery after device loss or failure. To address this problem, we propose an innovative failure recovery framework, which is realized through the IDC. More precisely, the IDC federates multiple independent factors (e.g., MC and BAA) that can be easily used in conjunction with a single secure backup password or real-world identity verification to reliably authenticate the user during recovery.

Fig. 4 depicts the proposed failure recovery mechanism. Initially, the user has to authenticate with the IDC using his secure backup password, which is required only for failure recovery. In case the user does not wish to maintain a backup password, he is able to access the IDC with real-world identity verification by scanning his e-Passport using his mobile device (step 1). If the backup password is correct, or the acquired identity matches the one that he had proved to the IDC before the failure, then he is granted only temporary and tentative access (AAL1) to the IDC, which provides limited functionality (step 2). In particular, the user cannot view, restore, or manage PABAC credentials and identity attributes. Conveniently, with tentative access the user is only able to view his trusted IdPs and initiate authentication to them, the AALs of each IdP, and the backup passwords for the BAAs and his FIDO or other AAL1 IdPs.

Subsequently, the IDC acts as an SP requesting from the user to authenticate with one of his trusted AAL3 IdPs, for example an MNO IdP via MC (step 3). Since the user cannot cannot use FIDO to authenticate, he is able to authenticate via SMS[2] using his newly issued SIM card (step 4). 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 MNO IdP that the given device was reported as lost and a new SIM card was issued (step 5).

Next, for increased assurance, the IDC needs to verify the behavior of the user through one of the trusted BAAs that are registered under his account (step 6). To do this, the user first authenticates to his BAA using a BAA-specific backup password (step 7). 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. The user is also able to backup all his BAA-specific backup password to the IDC, which are viewable in tentative mode. After the user has authenticated, the BAA grants him tentative access and he is not allowed to manage his behavioral profile until his signature is verified as that of the legitimate user's (step 8). With tentative 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 (step 9).

[2] We acknowledge the vulnerabilities of the SS7-based SMS system [18]. The authentication to the MNO IdP can also 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 [13].

Once the BAA has collected sufficient records to give a verdict on whether the user behaves as usual, the user is able to prove his behavior to the IDC (step 10). When he does so, the IDC acts as an SP while the BAA acts as an IdP authenticating the user based on his behavior and the result is returned to the IDC via OIDC (steps 11-15). If the verdict is negative the BAA locks that device out of its IdP. If the verdict is positive the user is granted full access (AAL3) to the IDC and he is able to issue new FIDO credentials for his account to the new device (step 16). Both MC and BAA authentication is needed because BAA does not formally increase the NIST authenticator assurance level.

F. De-anonymization Risks and Privacy Assessment

We extend OIDC so that it keeps a history of the identity attributes revealed to SPs. Using this information, we provide to the users privacy risk indicators that define the risk of involuntary de-anonymization. De-anonymization risk calculation is separated into two categories based on the protocol that a user is using to authenticate: 1) vanilla OIDC; and 2) PABAC. In the OIDC case, we calculate the probability with which an SP can infer the value of an attribute that the user has not explicitly revealed based on the attributes he 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 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.

G. Multi-device Support

We modify 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 maintain. This modification enables the users to maintain multiple devices. Besides this, a user is able to authenticate to an SP through his 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.

H. Deployability and Adoption

Our federated architecture have many significant benefits for adopters. First, user experience is enhanced since a user has to consolidate and prove his identity once at the IDC and then it can be re-used to access multiple IdPs and SPs. Second, there is a significant cost reduction to both the end-users (reduction in authenticators) and the SPs (reduction in infrastructure). Users do not have to remember dozens of passwords and at the same time they are able to retain their anonymity using PABAC, while SPs can offer FIDO and PABAC authentication to their end-users without the need to deploy any cryptographic credential stacks. There is also a 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.

Furthermore, it is clear that IdPs are crucial in federated architectures. However, what are the incentives for an organization to play the role of an IdP? By participating in our architecture, an IdP has many benefits. For example, an organization who maintains identity information about users (e.g., age) can offer age verification services to SPs that require age verification from their end-users to abide by the online age verification requirements imposed by regulators (i.e., the Gambling Act 2005[3] for remote gambling in UK).

VI. 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 V.

OIDC/FIDO UAF. To exploit the OIDC Provider features, we make use of the OpenAM software[4]. We implemented, within the OIDC Provider, a custom 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 a 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 API[5], which utilizes both underlying cryptographic protocol stacks used in our architecture.

Identity Consolidator. We have implemented the IDC and its respective modules as a Web application. Within the IDC we have 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. We have also implemented a custom module within the IDC that allows the IDC to act as an MC proxy. This custom module invokes a GSMA API Exchange-enabled [19] discovery service on a

[3] https://www.legislation.gov.uk/ukpga/2005/19/contents

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

[5] https://goo.gl/dkG5R8

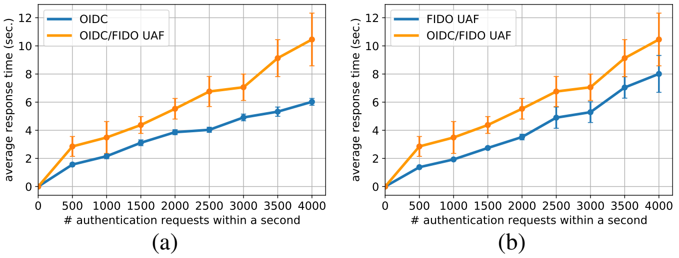

Fig. 5: Average response time of an OIDC/FIDO UAF authentication request compared with vanilla: (a) OIDC authentication request; and (b) FIDO UAF authentication request.

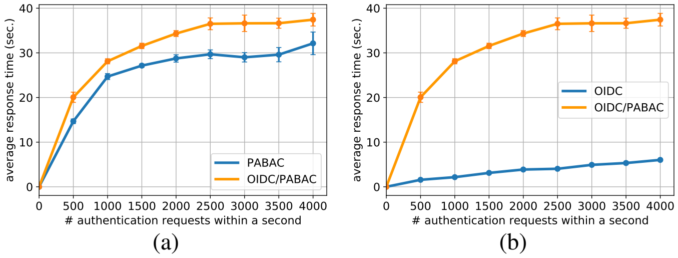

Fig. 6: Average response time of an OIDC/PABAC authentication request compared with vanilla: (a) Idemix authentication request; and (b) OIDC authentication request.

trusted MC Provider. This API is mainly used as the federation mechanism for MC authentication.

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

VII. Evaluation

In this section we evaluate our prototype implementation in terms of performance and security.

A. *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. For a more fair evaluation, when evaluating our federated PABAC authentication solution, we choose Idemix instead of U-Prove because it is the one with the lower performance (see Subsection V-C). Each experiment was conducted by sending a batch of authentication requests within a second starting from 500 to 4K 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 VI, 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 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 his password) and we use a 20 characters long password for authentication in the vanilla OIDC case. 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 and FIDO authentication, and we simulate the parallel authentication processes using different threads.

Following the same approach, we also evaluate our OIDC/FIDO UAF authentication module. Again, the simulations are 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 5(a) and 5(b) present the results of the evaluation of our OIDC/FIDO 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. We note that all the experiments were conducted using an OpenAM software stack and a FIDO UAF server that both run on the IdP and the client requests are executed using an Internet connection.

OIDC/PABAC. Next, we evaluate 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 6(a) and 6(b) 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 one of the vanilla Idemix.

B. 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 he 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 his 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 protocols 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 TrustZone, 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 he 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 his online accounts on that device using the AMM.

Service and Identity Providers. SPs and IdPs face various threats. Initially, by establishing protected sessions between the SPs/IdPs and the users, our solution guarantees that the access tokens are never exposed to unauthorized parties during an authentication. Next, 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. To also 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. Also, 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 preserve the users' privacy by employing advanced unlinkable and untraceable cryptographic credentials that are used to authenticate with PABAC-enabled IdPs. Additionally, using the Consent Management module of the IDC, a user has to provide his consent when revealing identity attributes to SPs. Last, the Profile Management module of the IDC offers to the users 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.

VIII. Related Work

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

A. Password alternatives

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, [3], [20], [21] 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 reuse the same password across multiple domains.

To overcome these issues, password managers like LastPass [11] and RoboForm [12] allow users to use a variety of strong passwords for accessing their online services, while the burden of maintaining and remembering them is offloaded to the password manager. However, some studies [22], [23] 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. Password managers are also 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.

Other studies propose alternatives to the password paradigm. Stajano [24] 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. Trusona [25] offers device-centric password-less and multi-factor authentication through a mobile application. A user can register by scanning one of his identity documents.

B. Identity Federation and Management

The past two decades numerous identity federation and management solutions have emerged. WSO2 Identity Server [26] 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 [27] 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 Alliance [28] 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. [29] propose a framework that uses information about identities that is aggregated across multiple domains to reason about their trustworthiness. Instead, we deploy more sophisticated algorithms for assessing the trustworthiness of a user's identity with high confidence (see Subsection IV-B).

C. 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 [30], [31]. Others [32]–[35] continuously track users' behavior for authentication purposes based on various behavior types. In most of these approaches, the classifier's location is not specified and they do not consider battery, computational, and space limitations. At the same time, they only tackle observation and impersonation attacks. Song et al. [36] propose TrustCube, a framework that leverages federated authentication schemes to authenticate users based on their behavior on behalf of any SP. BehavioSec [37] offers continuous behavioral authentication software as a service. It uses 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 architecture, BAAs are offered as independent entities that can harvest user behavior data from a user's device in a non-intrusive and battery efficient way. More precisely, we propose an open architecture under which any entity able to capture behavior can offer behavioral authentication via OIDC, while also offering enhanced protection against attackers that manage to compromise the device.

D. 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 model uses either one of two attribute based encryption (ABE) methods. Key-policy ABE [38] uses the policies to create the applicant keys and uses the attributes to describe the encrypted data. Ciphertext-policy ABE [39] uses a tree form access policy, where attributes are the leaves of the tree. Ruj et al. [40] propose a privacy-preserving access control scheme, in which the attributes of each user belong to multiple key distribution centers [41]. The user's identity information is stored in the cloud, which acts as the verifier for the users' credentials. However, user privacy is not protected in the cloud. Chase [42] 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. Later, Chase and Chow [43] 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, in our architecture we integrate cryptographic credentials stacks (Idemix [8] and U-Prove [9]) to let users prove their identity attributes to SPs using cryptographic credentials that are securely stored on their devices. 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.

IX. 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 received funding from the European Union's Horizon 2020 Framework Program under the ReCRED project (GA No. 653417), and the Marie Skłodowska-Curie INCOGNITO project (GA No. 824015).

References

[1] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 553–567.

[2] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021–2040, 2003.

[3] A. Adams and M. A. Sasse, "Users are not the enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 41–46, 1999.

[4] D. Florencio and C. Herley, "A large-scale study of web password habits," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 657–666.

[5] B. Ives, K. R. Walsh, and H. Schneider, "The domino effect of password reuse," *Communications of the ACM*, vol. 47, no. 4, pp. 75–78, 2004.

[6] OpenID Foundation, "OpenID Connect," http://openid.net/connect, 2014.

[7] FIDO Alliance, "FIDO UAF Specification," https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-uaf-overview-v1.1-id-20170202.html, 2017.

[8] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 21–30.

[9] C. Paquin and G. Zaverucha, "U-prove cryptographic specification," https://www.microsoft.com/en-us/research/publication/u-prove-cryptographic-specification-v1-1-revision-3/, 2013.

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

[11] LogMeln, Inc., "LastPass: # Password Manager & Vault App," https://www.lastpass.com, 2019.

[12] Siber Systems, Inc., "RoboForm: Manage your passwords with ease and security," https://www.roboform.com/, 2019.

[13] GSMA, "Introducing mobile connect – the new standard in digital authentication," https://www.gsma.com/identity/mobile-connect, 2019.

[14] N. I. of Standards and Technology, "Nist - digital identity guidelines," https://pages.nist.gov/800-63-3, 2017.

[15] B. McGillion, T. Dettenborn, T. Nyman, and N. Asokan, "Open-tee–an open virtual trusted execution environment," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1. IEEE, 2015, pp. 400–407.

[16] O. 2.0, "Industry-standard protocol for authorization," https://oauth.net/2/, 2019.

[17] F. Corella and K. Lewison, "Privacy postures of authentication technologies," in *The Internet Identity Workshop (IIW)*, 2013.

[18] D. Hugoo, "SS7 SMS-Based Exploit: A Wake-Up Call to Shift to Stronger Two-Factor Authentication," https://blog.easysol.net/ss7_sms_based_exploits/, 2017.

[19] GSMA, "API Exchange: Global API Federation capability," https://www.gsma.com/identity/api-exchange, 2019.

[20] R. Morris and K. Thompson, "Password security: A case history," *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[21] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password memorability and security: Empirical results," *IEEE Security & privacy*, vol. 2, no. 5, pp. 25–31, 2004.

[22] S. Chiasson, P. C. van Oorschot, and R. Biddle, "A usability study and critique of two password managers." in *USENIX Security Symposium*, vol. 15, 2006, pp. 1–16.

[23] R. Zhao, C. Yue, and K. Sun, "Vulnerability and risk analysis of two commercial browser and cloud based password managers," *ASE Science Journal*, vol. 1, no. 4, pp. 1–15, 2013.

[24] F. Stajano, "Pico: No more passwords!" in *International Workshop on Security Protocols*. Springer, 2011, pp. 49–81.

[25] Trusona, "Passwordless authentication," https://www.trusona.com, 2015.

[26] WSO2, "Identity & access management," http://wso2.com/identity-and-access-management, 2019.

[27] D. Recordon and D. Reed, "Openid 2.0: a platform for user-centric identity management," in *Proceedings of the second ACM workshop on Digital identity management*. ACM, 2006, pp. 11–16.

[28] Liberty Alliance, "The Liberty Alliance Project," http://www.projectliberty.org, 2019.

[29] 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*. International World Wide Web Conferences Steering Committee, 2016, pp. 1249–1259.

[30] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors," in *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, 2014, pp. 221–232.

[31] 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 *proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 987–996.

[32] Z. Jorgensen and T. Yu, "On mouse dynamics as a behavioral biometric for authentication," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, 2011, pp. 476–482.

[33] 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 Conference on Technologies for Homeland Security (HST)*. IEEE, 2012, pp. 451–456.

[34] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit authentication through learning user behavior," in *International Conference on Information Security*. Springer, 2010, pp. 99–113.

[35] M. Jakobsson, E. Shi, P. Golle, and R. Chow, "Implicit authentication for mobile devices," in *Proceedings of the 4th USENIX conference on Hot topics in security*, 2009, pp. 9–9.

[36] Z. Song, J. Molina, S. Lee, H. Lee, S. Kotani, and R. Masuoka, "Trustcube: An infrastructure that builds trust in client," in *Future of Trust in Computing*. Springer, 2009, pp. 68–79.

[37] BehavioSec, "Continuous authentication through behavioral biometrics," https://www.behaviosec.com, 2018.

[38] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. Acm, 2006, pp. 89–98.

[39] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 2007, pp. 321–334.

[40] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy preserving access control with authentication for securing data in clouds," in *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. IEEE, 2012, pp. 556–563.

[41] P. D'Arco and D. R. Stinson, "On unconditionally secure robust distributed key distribution centers," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2002, pp. 346–363.

[42] M. Chase, "Multi-authority attribute based encryption," in *Theory of cryptography conference*. Springer, 2007, pp. 515–534.

[43] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 121–130.