ResearchArticle Decentralized, Privacy-Preserving, Single Sign-On

Omid Mir ,1 Michael Roland ,2 and Ren´ e Mayrhofer 2 1Johannes Kepler University Linz, LIT Secure and Correct Systems Lab, Linz, Austria 2Johannes Kepler University Linz, Institute of Networks and Security, Linz, Austria Correspondence should be addressed to Omid Mir; mir@ins.jku.at Received 2 April 2021; Accepted 11 November 2021; Published 22 January 2022 Academic Editor: David Meghias Incurrentsinglesign-onauthenticationschemesontheweb,usersarerequiredtointeractwithidentityproviderssecurelytoset upauthentication dataduringaregistration phaseand receivea token(credential)forfutureaccesstoservicesand applications. This type of interaction can make authentication schemes challenging in terms of security and availability. From a security perspective,amainthreatistheftofauthenticationreferencedatastoredwithidentityproviders.Anadversarycouldeasilyabuse such datatomountanofflinedictionaryattackforobtainingtheunderlying passwordorbiometric. Fromaprivacyperspective, identityprovidersareabletotrackuseractivityandcontrolsensitiveuserdata.Intermsofavailability,usersrelyontrustedthirdpartyserversthatneedtobeavailableduringauthentication.Weproposeanoveldecentralizedprivacy-preservingsinglesign-on scheme through the Decentralized Anonymous Multi-Factor Authentication (DAMFA), a new authentication scheme where identity providers no longer require sensitive user data and can no longer track individual user activity. Moreover, our protocol eliminates dependence on an always-on identity provider during user authentication, allowing service providers to authenticate users at any time without interacting with the identity provider. Our approach builds on threshold oblivious pseudorandom functions(TOPRF)toimproveresistanceagainstofflineattacksandusesadistributedtransactionledgertoimproveavailability. WeprovethesecurityofDAMFAintheuniversalcomposibility(UC)modelbydefiningaUCdefinition(idealfunctionality)for DAMFA and formallyproving the security of our scheme via ideal-real simulation. Finally, we demonstrate the practicability of our proposed scheme through a prototype implementation.

1.Introduction Authenticated Key Exchange (AKE) is one of the most broadly used cryptographic primitives that enable two parties to create a shared key over a public network. Typically, the parties need to have authentication tokens, e.g., cryptographickeys (asymmetricorsymmetrichigh-entropy keys) or short secret values (low-entropy passwords). They also securely store these authentication tokens in a trusted service provider during the registration phase. There are various types of authentication factors such as knowledge, possession, and physical presence; low-entropy passwords are widely present in practice. An example of an authenticationprotocolthatreliesonpasswordsisPassword-Based Authenticated Key Exchange (PAKE) [1]. However, passwords are usually vulnerable to both online and offline attacks [2, 3]. An attacker who

compromisesthedatastoredwiththeserviceprovider(user account data, consisting of usernames and associated (potentially salted) password hashes) can run an offline dictionary attack on that data. Such an attack leads to the disclosure of user accounts and this has happened several timesinthepast,cf.[2,4,5].Eveniflow-entropypasswords are correctly salted and hashed, they still do not resist the bruteforceofmodernhardware.Alreadyin2012,arigof25 GPUs could test up to 350 billion guesses per second in an offline dictionary attack [6]. Multi-Factor Authentication (MFA) schemes overcome this risk by adding additional authentication factors. MFA combines (low-entropy) passwords with, e.g., secret values stored in physical tokens. Recent advancements in fingerprint readers and other sensors have led to the increased usage of smartphones and biometric factors in MFA schemes (e.g, the use of biometrics to securely retrieve

Hindawi Security and Communication Networks Volume 2022, Article ID 9983995, 18 pages https://doi.org/10.1155/2022/9983995

private information [8]) Figure 1, although these methods make the guessing of authentication factors more difficult. However, some MFA schemes incorporate password authentication and second-factor authentication as separate mechanismsandstoreasaltedpasswordhash(orbiometric) on the server, leading to different vulnerabilities such as spoofing and offline attacks [7, 9]. In other words, an adversary compromising the server is still able to recover the actual password (even if that password is no longer usable withouttheadditionalassociatedfactors).Moreover,mobile devices (smartphones, wearables, FIDO U2F, etc.) are considered more likely to be subject to loss or theft, and particularly smartphones and wearables open a large, highrisk attack surface for malware [10, 11]. In general, authentication schemes are designed to uniquely identify a user. Consequently, they do not aim at protecting user privacy, and users' activity in the digital world can easily be logged and analyzed. Leakage of individualinformationmayhaveseriousconsequencesforusers (including financial losses). To meet the increasing need of privacy protection in the digital world, multi-factor authentications are enhanced with privacy-preserving technologies. For instance, anonymous authentication schemes allow a member of a legitimate group, called a prover, to convince a verifier that it is a member of the group without revealing any information that would uniquely identify the prover within the group. Various schemes for anonymous password authentication have been proposed, e.g., [12–15]. Inparticular,anonymouspasswordauthenticationpromises unlinkability: The prover (e.g., the server of a service or identity provider) should not be able to link user authentications.Therefore,foranytwoauthenticationsessions,the proverisunabletodetermineiftheyhavebeenperformedby the same user or two different users.

1.1. Building a Fully Decentralized Authentication Architecture. An Identity Provider (IDP) with a centralized database of authentication data of all users could easily provideanMFAschemeandofferconvenientsinglesign-on (SSO)tootherservicesforitsusers[16].SSOallowsusersto once receive a single token (identity) provided by IDP and repeatedly authenticate themselves to servics providers. Several initiatives such as PRIMA [17], OAuth [18], SAML [19],andOpenID[16]letserviceproviderstakeadvantageof another centralized identity provider to authenticate users without becoming responsible for managing account passwords. In all these systems, the authentication follows a similar scheme (see Figure 2) [20]: (1) Intheregistrationphase,theusercreatescredentials (e.g., a username/ID and a password) and passes them to the IDP (a trusted server) which stores the username together with the hash of the password. (2) Intheauthenticationphase,theIDPverifiestheusersupplied sign-on credential by matching the username and password hash. (3) After successful verification, the IDP issues an authentication credential (a digital signature or a

message authentication code) using a master secret keythatauthenticatestheusertotheserviceprovider (e.g., a website) they want to visit. However, this kind of centralized system poses several challenges: (1) The IDP represents a single point of failure and an obvious target for attacks, such as: (a)extractionofthesecretkeytoforgetokens,which enable access to arbitrary services and data in the system; (b) capturing hashed passwords (or biometrics) to run offline dictionary attacks in order to recoverusercredentials,bothpotentiallyresultingin severe damage to the reliability of the system [20]. (2) The IDP is actively involved in each authentication sessionandcan,therefore,trackuseractivity,leading to serious privacy issues [21, 22]. (3) The IDP takes a significant amount of control over thedigitalidentityawayfromtheuser.Userscannot fully manage and store their identity by themselves but always need to rely on and interact with an available IDP that offers the identity management system to them and the service provers they want to interact with (active verification).

1.2.OurContribution. To address the above challenges, we construct a novel decentralized privacy-preserving single sign-on scheme using a new Decentralized Anonymous Multi-Factor Authentication (DAMFA) scheme, where the processofuserauthenticationnolongerdependsonasingle trusted third party. Instead, it is fully decentralized onto a sharedledgertopreserveuserprivacywhilemaintainingthe singlesign-onproperty.Thatis,usersdonotneedtoregister theircredentialswitheachserviceproviderindividually.The scheme also permits services where authenticating users remain anonymous within a group of users. Subsequently, ourschemedoesnotrequiretheIDPtobeonlineduringthe verification (passive verification). Moreover, since there is no single third party (i.e., the IDP) in control of the whole authenticationprocess,userandusagetrackingbytheIDPis inhibited. The passive verification property of our scheme allows service providers to authenticate users at any time without requiring additional interaction with an IDP except what is available on the shared ledger. This property removes the costofrunningsecurechannelsbetweentheserviceprovider and the identity provider. Simultaneously, the IDP is eliminated as a single point of failure and attack within the authentication process. Theschemereliesonpersonalidentityagentsasauxiliary devicesthatassisttheuserintheauthenticationprocess.The personal identity agents participate in a threshold secret sharing scheme to store the distributed private key of their users. In the authentication phase, the user unlocks their private key through a combination of biometrics and a password,combiningbiometric, knowledge,andpossession factors. The distributed architecture prevents offline attacks

against data extracted from compromised agents, as long as only a set of agents below the threshold is compromised or corrupted. We define the ideal functionality and real-world definitions for the security of our DAMFA scheme. We prove our construction’s security via ideal-real simulation, showing the impossibility of offline dictionary attacks. Finally, we demonstrate that our protocol is efficient and practical through a prototypical implementation and throughacomparisonofourschemewithotherSSOworks.

2.Related Work 2.1. Single-Factor (Password) Authentication Key Exchange. For a long time, knowledge was (and still is) used as a primary means of authentication. Single-factor authentication based on passwords and PINs is a mechanism that is well-studied. Bellovin and Merritt [24] proposed Encrypted Key Exchange (EKE) where a client and a server share a password and use it to exchange encrypted information to agreeonacommonsessionkey.EKEwasfollowedbyseveral enhancements (cf. [25–27]). Bellare et al. [1] expanded this to a general formal provable model for Password Authentication Key Exchange (PAKE). After that, two generic schemes of PAKE were proposed by Gennaro and Lindell [28]and by Groceand Katz[29] whichare among themost efficient ways of constructing PAKE in the standard model. BenhamoudaandPointcheval[30]explicitlyintroducea verifierintotheauthenticatedkeyexchange,whereaverifier isahashvalueortransformation V � H(s,pw) ofthesecret

password pw with a public salt s, and the server stores the pair (s,V) for each user.

2.2.Multi-FactorAuthentication. Asingleknowledge-based authenticationfactorhasthedisadvantagethatanadversary needs to only compromise that single factor. Multi-factor authentication (MFA) overcomes this by combining multiple different factors. The widely used combination is longterm passwords with secret keys, possibly stored in tokens (e.g., FIDO U2F). Shirvanian et al. [31] introduce a framework to analyze such two-factor authentication protocols. In their framework, the participants are a user, a client (e.g., a web browser), a server, and a device (e.g., a smartphone). In the authentication phase, the user sends a password and some additional information provided by the device. In most existing solutions, including Refs. [31–33], during the registration process, the user gets a value called the "token," while the server records a hashed password. During the authentication phase, the two required factors (the password and the token) are sent to a verifier. Jarecki et al. [34] provide a device-enhanced passwordauthenticated key exchange protocol employing mobile device storage as a token. This setting serves two purposes: Firstly, for an adversary to successfully mount an offline dictionary attack, they must corrupt the login server in addition to the mobile device storage. Secondly, the user must confirm access to the mobile device storage during login. Another popular factor used to authenticate users to remote servers is biometrics [35–38]. Fleischhacker et al. [39] also propose a modular framework called MFAKE which models biometrics following the liveness assumption ofPointchevalandZimmer[37].However,Zhangetal.[40] demonstrate that their scheme does not adequately protect privacy. Indeed, biometric authentication becomes a weak point when the framework directly uses the biometric template for authentication. In addition, it requires to, respectively, execute a lot of sub-protocols which makes the scheme inefficient.

2.3.AnonymousAuthentication. Anotherapproachtowards user authentication is the anonymous password authenticationprotocolproposedbyVietetal.[12].Theycombinean oblivious transfer protocol and a password-authenticated keyexchangescheme.Furtherenhancementswereproposed by Refs. [14, 15, 38]. Ananonymousauthenticationprotocolpermitsusersto authenticatethemselveswithoutdisclosingtheiridentityand

Multi-factor authentication

Biometric factor: Fingerprint, face recognition, behavior recognition

Two-factor authentication

Ownership factor: Smartphone, key-card one-time password

Single-factor authentication

****

Knowledge factor: PIN, password, security questions Figure 1: Evolution of authentication methods from SFA to MFA [7].

Identity provider

Service provider

(3). Token/Fail (2). If (usr, h) & (usr, h*) Matches then generate a token

(4). Token

User

(5). Succ/ Fail

(1). {usr, $h^* = H(pwd^*)$}

Figure2:Thegenericflowdiagramshowstheauthenticationphase ofapassword-basedtokenmethod.Thefiguredoesnotincludethe registration phase where the users store their username (usr) and hashed password (h) with the identity provider.

becomes an important method for constructing privacypreserving authenticated public channels. Zhang et al. [40] presented a new anonymous authentication protocol that relies on a fuzzy extractor. They consider a practical application and suggest several authentication factors such as passwords, biometrics (e.g., fingerprint), and hardware with reasonably secure storage (e.g., smartphone).

2.4.SummaryofRelatedWorks. Single-factorauthentication based on passwords is a primary means of many authenticationprotocols[1,25,28,41].Multi-factorauthentication (MFA) overcomes the problem of compromise in a single factor by combining multiple different factors [31, 34, 40, 42, 43]. An anonymous authentication protocol permits users to authenticate themselves without disclosing their identity [12, 14, 15, 44]. Finally, SSO allows users to once receive a single token provided by IDP and repeatedly authenticate themselves to service providers [16, 17, 19, 45, 46].

3.Building Blocks 3.1. Pointcheval and Sanders Signature. Our work relies on thecredentialsschemeproposedbyPointchevalandSanders [47]. The scheme works in a bilinear group ($G1$,$G2$,GT) of type 3, with a bilinear map e: $G1 \times G2 \longrightarrow GT$ and has the following algorithms: (1) Setup(1λ)⟶(params): Choose a bilinear group ($G1$,$G2$,GT) with order p, where p is a prime number. Let g1 be a generator of G1, and g2 a generator of G2. The system parameters are params � G1,G2,GT,p,g1,g2 ⍰. (1)(2) KeyGen (params)⟶(sk,vk): Choose a randomsecret key sk � (x,y)∈Zp. Parse params, and publish the verification key vk � g2,X,Y ⍰ � g2,gx 2,gy 2 ⍰. (2)(3) Sign (params,sk,m)⟶(σ):Parse sk � (x,y).Pick a random element h∈G1, and output σ �(h,s) � h,hx+y·m ⍰. (3)(4) Verify (pk,m,σ): Parse σ as (σ1,σ2) and check whether σ1≠1G1 and e(σ1,X · Ym) � e(σ2,g2) are both satisfied. In the positive case, it outputs 1, otherwise 0. The signature σ � (h,s) is randomizable by choosing a random r′∈Zp and computing σ′� (hr′,sr′). The above scheme can be modified to obtain a signature on a hidden message (commitment) and also offers a protocol to show a zero-knowledge proof of a signature σ � (σ1,σ2).

3.2.ObliviousPseudo-randomFunction(OPRF). A pseudorandom function (PRF) F is a function that takes two

inputs: a secret function key k and a value x to compute on. It outputs Fk(x), a function picked randomly from a PRF family, which is secure if it is distinguishable from a random function with the same domain and range with a negligibleprobabilityforallprobabilisticpolynomialtime (PPT)distinguishers.AnobliviousPRF(OPRF,cf.[48])is a protocol between two parties (a sender and a receiver) that securely computes Fk(x) where both x and k are the inputs of sender and receiver, respectively, such that no party learns anything except for the input holder that learns Fk(x). A threshold OPRF (TOPRF, cf. [49]) is an extension of the OPRF which allows a group of servers to secret share a key k for a PRF F with a shared PRF evaluation protocol which lets the user compute Fk(x) on an input x, so that both x and k are secret if no more than t of n servers are corrupted (see Figure 3). A formal definition of the TOPRF protocol as a realizationoftheTOPRFfunctionalityisgiveninFigure4.Note that we just duplicate these functionalities so that readers can easily follow our ideal functionality and construction (for more details see [49]).

3.3.SecretSharingScheme. Asecretsharingschemeconsists oftwoPPTalgorithms[50]:First,TSSGengeneratesnshares of the secret key K as ⟨k1,...,kn⟩⟵TSSGen(K), and secondTSSReconuses t sharestoretrievetheprimarysecret value K as K⟵TSSRecon(s1,...,st). The security assumptionofthisschemeisthatanyamountofsharesbelow thethresholddoesnotdiscloseanyinfoaboutthesecretkey.

3.4.PublicAppend-OnlyLedger. Aledgerallowsustokeepa list of public information and maintains the integrity of the dataset.Itguaranteesaconsistentviewoftheledgerforevery party.Everyusercaninsertinformationintotheledgerand, once some data are uploaded, nobody can delete or modify it. Moreover, the ledger assures the correctness of pseudonyms and guarantees that no one can impersonate another participant to release information. Furthermore, it distributes up-to-date data to all participants. In this paper, we assume this assumption holds and construct our system on theblockchain technique as apublic append-only ledger (blockchain). There are already some works constructing advanced applications based on this assumption, such as Refs. [51–53]. Yang et al. [54] formally define a public append-only ledger, which we use for constructing our DAMFA system (see Figure 5). FB executes the following steps with parties PA1,...,PAn⍰ ⍰ and an ideal adversary S as follows: (1) Initialize. Initialize creates an empty list Lp in the beginning. (2) Store. On input (Store,PAi,Nymo u,M), checks that Nymo u is a valid pseudonym for PAi, then stores the tuple (Nymo u,M) to Lp anddeclaresto S thatanew item was appended to the list Lp.

(3) Retrieve.Oninput(Retrieve,PAi),returnsthelistLp to PAi.

3.5. Zero-Knowledge Proof of Knowledge. In a zeroknowledge proof of knowledge system [55], a prover

proves to a verifier that it possesses the witness for a statement without revealing any additional information. In this paper, we use noninteractive zero-knowledge proofs known as Fiat-Shamir heuristic [56] as they have the advantage of being noninteractive. For example, NIZKPoK denotes a noninteractive zero-knowledge proof of the elements x and y as NIZKPoK (x,y) ⍰ : h � gx∧c � gy} that satisfies both h � gx and c � gy. Values (x,y) are assumed to be hidden from the verifier. Similarly,thealgorithmcanadmitamessageasinput,thusit is also called signature proof of knowledge denoted as ZKSoK[m] (x,y): ⍰ h � gx∧c � gy}. 3.6. Dynamic Accumulators. A dynamic accumulator is a primitive allowing a large set of values to be accumulated intoasinglequantity,theaccumulator.Foreachvalue,there existsawitnesswhichistheevidenceattestingthatthevalue is indeed contained in the accumulator. The proof of showing that a value is part of an accumulator can be zeroknowledge proof, which reveals neither the value nor the witnesstotheverifier.Camenischetal.[57]defineaconcrete construction of dynamic accumulators with the five algorithms AccSetup, AccAdd, AccUpdate, AccWitUpdate, and AccVerify: (1) AccSetup: This is the algorithm to output the public parameters. Select bilinear groups paramsBM � (q, G,GT,e,g) with a prime order p and a bilinear map e. Select g∈G. Select c∈Zp. Generate a key pair msk and pk forasecuresignaturescheme.Compute and publish p,G,T,e,g, ⍰ g1 � gc1,..., gn � gcn, gn+2 � gcn+2,...,g2n � gc2n}andz � e(g,g)cn+1 asthe public parameters. (2) AccAdd (skA,i,accV,stateU). Compute ω � ⍰j≠i j∈Vg n+1−j+i andasignatureσi ongi ‖ iundersigningkey sk. The algorithm outputs witi � (ω,σi,gi), an updated accumulator value accVUi � accV · gn+1−i, and stateUUi � (UUi { },g1,... ,gn,gn+2,...,g2n). (3) AccUpdate: This is the algorithm to compute the accumulator using the public parameters. The accumulator accV of V is computed as accV � ⍰ i∈V gn+1−i, (4) (4) AccWitUpdate:Thisisthealgorithmtocomputethe witness that values are included in an accumulator, using the public parameters. Given V and the accumulatoraccV,thewitnessofvaluesi1,...,ik inUis computed as ω′� ω · ⍰j∈V/Vωgn+1−j+i ⍰j∈Vw/Vgn+1−j+i. (5) (5) AccVerify:This is thealgorithm to verify thatvalues in U are included in an accumulator, using the witness and the public parameters. Given accV, stateU, and ω, accept if

Figure 3: (n,t)-threshold computation ina TOPRF protocol[49].

Figure 4: Functionality FTOPRF [49].

Figure 5: Functionality FB [54].

e gi,accV ⍰ e(g,ω)

� z. (6) As Camenisch et al. [57] point out, the purpose of an accumulator is to have accumulator and witnesses of size independent of the number of accumulated elements.

3.7. Pedersen Commitments. Using a commitment scheme, users can bind themselves to a chosen value without revealing the actual value to a third party receiving the commitment. Thereby, a user cannot change their choice (binding), and, at the same time, the recipient of a commitment does not learn anything about the actual value the user committed to (hiding of the value). Pedersen commitments [58] have a group G of prime order q and generators (g0,...,gm) as public parameters. For committing tothevalue (z1,...,zm)∈Zq,auserpicksarandom r∈Zq and sets C � PedCom(z1,...,zm; r) � gr 0 ⍰m i�1 gzi i . 4.Decentralized Anonymous Multi-Factor Authentication (DAMFA) We build a new practical decentralized multi-factor authentication scheme, Decentralized Anonymous MultiFactorAuthentication (DAMFA),where theprocessofuser authentication no longer depends on a single trusted third party.Theschemealsopermitsserviceswhereauthenticating users remain anonymous within a group of users. Subsequently, our scheme does not require the IDP to be online during the verification. To protect the private key of their user,weusepersonalidentityagentsasauxiliarydevicesthat participate in a threshold secret sharing scheme to store the distributed private key of the user.

4.1.SystemModel. The overall system model of DAMFA is shown in Figure 6. The protocol is executed between four participants: (1) User U: A user who wants to access various services offered by different service providers. During the registrationphase(whichrunsonlyonce), U obtains abiometrictemplateBiofromasensorandchoosesa password pw. In the authentication phase, users U interact with a set of personal identity agents to authenticate themselves in an anonymous manner. (2) Personal identity agent PAi: We associate each user with a set of personal agents which are auxiliary devices that assist a user in creating a credential for authentication. These personal agents remain under the administrative control of their associated users, who can freely choose where to run them. For example,theycouldberunonasmarthomecontroller, at a cloud provider, or even on a mobile phone. U generatesaprivatekeyandexecutesthresholdsecret sharingontheprivatekeytogeneratesecretsharesof that private key. The user stores the secret shares

among their personal agents such that each PAi has one share of the overall secret key. (3) Service provider (verifier) SP: These are the service providers (untrusted and distributed servers) that requireauthenticationfromauser U.Afterverifying a user’s credentials, they provide access to the corresponding service. (4) Identity provider IDP: The identity provider is an entity that issues credentials to users. These credentials grant permission to use specific services by proving membership of a specific permission group (clients, employees, department members, account holders, subscribed users, etc.). In addition, users act as nodes in the blockchain network: Theycollaborativelymaintainalist of credentialsina public ledger (blockchain) and enforce a specific credential issuing policy when adding to that list. For more details on howthesestepswork,werefertosubsection4.3.,High-Level View.

4.2. Threat Model. In order to demonstrate the security of the proposed protocol, we determine the capabilities and possible actions of an attacker. We consider a PPTattacker who has perfect control of the communication channels. Theycaneavesdropallmessagesinpublicchannelsandalso modify, add, and remove messages on the network. The attackercan,atanytime,corrupt (t − 1) oftheuser'sagents (nomorethanthresholdt),inwhichcasetheattackerknows all the long-term secrets (such as private keys or master shared keys). In the proposed protocol, we consider some privacy requirements such as unlinkability, identity privacy, and user data privacy: Unlinkability means that an adversary cannot distinguish a user who is authenticating from any (other) user who has authenticated in the past. Identity privacymeansthatanadversarycannotdetermineifagiven authentication credential belongs to a specific user. User data privacymeans thatanadversarycannot learn anything abouttheuser'ssensitiveauthenticationdata(i.e.,biometric data, password).

4.3. High-Level View. To build a fully decentralized authentication architecture, we need to set up a small distributed shared database (to store credentials) between nodes.Dataarehighlyavailable,butnobodyhascontrolover the database. Furthermore, users would never want to modifydatainthepast.Userdataneedtobeimmutable,and data should be publicly accessible. We employ a public append-only ledger in order to fulfill our requirements. A ledger (blockchain) maintains the integrity of the dataset andguaranteesaconsistentviewofthedataforeveryparty. Everyparticipantcanappendinformationtotheledgerand, once uploaded, nobody can delete or modify the data.

Definition 1 (DAMFA). A DAMFA system consists of a global transaction ledger instead of a single party

representing the organization. Moreover, the DAMFA scheme consists of the following phases: (1) Setup: In the setup phase, we define the public parameters and execute the following algorithm: U generatesaprivatekeyandexecutesthresholdsecret sharing(TSS)ontheprivatekeytogeneratesharesof that secret. The user stores the secret shares among their personal agents (similar to the initialization of TOPRF [49], done via a distributed key generation for discrete-log-based systems, e.g. Ref. [59]). (2) Registration: In the registration phase, the user U first selects a password pw and collects their biometric Bio at a sensor. Then, U runs the TOPRF protocol by interacting with personal agents to reconstruct the TOPRF secret key. After that, the IDP issuesamembershipcredentialthatshowsthatUisa validmember(employee,accountholder,subscribed user,etc.).Forthispurpose, U sendsarequestwitha pseudonym and a (noninteractive) zero-knowledge proof (NIZK) which indicates they are the owner of the pseudonym (they know the secret key that belongs to the pseudonym) and authenticate themselves to the IDP. Then, U receives a membership credential,whichisasignatureontheirpseudonym. The user U creates a pseudonym Nymo u and verification information, namely, a protected credential PCi, by encrypting the membership credential with the TOPRF secret key. Subsequently, U computes a NIZK proof that (1) the credential PCi and the pseudonym contain the same secret key and (2) proof of knowledge of the signature which is issued by the ID provider (i.e., she has valid group membership).Notethattheusercanexecutetheseactions in an offline state because no interaction with the public ledger is required. Finally, nodes accept the credential to the ledger if and only if this proof is valid. (3) Authentication: The user U attempts to access the services of an SP in an anonymous and unlinkable way. SP authenticatestheuserifandonlyif theuser provides a valid credential. First, a service provider sends an authentication request (which is a signature)toU.Theuserinsertsthepasswordpw∗andthe biometric Bio∗and runs the TOPRF protocol by interacting with personal agents to reconstruct the TOPRFsecretvalue.Ufirstscansthepublicledgerto obtain the accumulator AC, which is a set PC ��→ � PC1,...,PCn consisting of all credentials belonging to a specific IDP. Then, U finds their own protected credential PC∗ i within this set (via the pseudonym Nymo u). U decrypts PC∗ i using the TOPRF secret key and recovers the initial credential (a signature from IDP). U presents the credential under a different pseudonym Nymv u by proving in zero-knowledge that (1) they know a credential PCi on the ledger from IDP, (2) the credential opens to the same secret key as their own pseudonym Nymv u, and (3) they prove possession of a membership credentialfromIDP(thesignature),cf.[52].SPscans the public ledger to obtain the accumulator AC which is a set PC ��→ � PC1,...,PCn consisting of all credentials belonging to a specific organization.

Personal agents

1. Request

2. Response

3. Credential

ID Provider

User

6. Protected credential

4. Request membership credential

7. Upload credential 10. Collect data

8. Login request

9. Authentication message

11. Accept/reject

3. Recovery secret key

SP

Figure 6: A system model of the DAMFA scheme.

Then, it checks the validity of the candidate credential by finding the candidate credential in the set PC∗ i∈PC ��→ and checking proof of knowledge on the credential and pseudonym. 4.4. The DAMFA Functionality. We formally define the proposed scheme's security by presenting its ideal functionalitythatisimplementedviaatrustedpartyFTOPRF with apublicledger.Allcommunicationtakes placethroughthis idealtrustedparty.IntheUCframework[60,61],theremay besomecopiesoftheidealfunctionalityrunninginparallel. Each one is supposed to have a unique session identifier (SID). Each time a message is sent to a specific copy of functionality,suchthatthismessagecontainstheSIDofthe copy that is intended for. As noted in Ref. [49], we also use theticketingmechanism,whichensuresthatinordertotest a password and biometric guess, the attacker must impersonate t + 1 agents. To this end, they define a counter tx(p,PAi)foreachPAi∈SIinwhichtheparameterpisalso used to identify it. In addition, when an agent PAi∈SI completes its interaction, the functionality increases the counter tx(p,PAi). On the other hand, when a user, either honestorcorrupt,completesaninteractionthatisassociated to PAi, tx(p,PAi) decreases by 1. It ensures that for any honest agent PAi, the number of user-completed OPRF evaluations with PAi is no more than the number of agentcompleted OPRF evaluations of PAi. It sets t + 1 agent tickets for accessing the proper TOPRF result by reducing (nonzero) ticket counters tx(p,PAi) for an arbitrary set of t + 1 agents in SI. The ideal functionality as:

4.4.1. Registration (i) Uponreceiving (Reg,sid,SI,pw,Bio) for |SI| � PAn from U, records this message and sends (Reg,U,sid,SI) to A∗(Ignores other Reg cmd). Computes a secret key K using TOPRF protocol FTOPRF and if |SI∩CorrSrv|≥t + 1 then sends (K,pw,Bio) to A∗. (ii) Upon receiving (SReg,sid,PAi) from A∗, if a record ⟨Reg,U,sid,SI,pw,Bio⟩ exists and PAi∈SI then marks PA as active and sends (SInit,sid) to PA. (iii) Upon receiving (UReg,sid) from A∗, if the record ⟨Reg,U,sid,SI,pw,Bio⟩ exists and all agents in SI aremarkedactive,thenrunsacommitmentscheme FCom and an encryption FEnc to get (τi,ci), respectively, and sets the pseudonym as Nymo u � τi and PCi � ci as the credential. It records ⟨Nymo u,PCi,U,SI,K⟩, sends (sid,Nymo u,PCi) and (RegComplete,sid,SI) to its public ledger and A∗, respectively.

4.4.2. Authentication (i) Upon receiving (Auth,sid,ssid,SR,pw′,Bio′) for |SR| � t + 1 from U∗, retrieves ⟨Reg,U,sid,SI, pw,Bio,K⟩, records ⟨Auth,U∗,sid,SI,SR,pw,pw′,

Bio,Bio′⟩, and sends (Auth,U∗,sid,ssid,SR) to A∗. IgnoresfutureAuthcommandsinvolvingthesamessid. (ii) Upon receiving (SAuth,sid,ssid,PAi) from A∗, if PAi∈SR is marked active then sets tx(PAi) + + (sets it to 1 if it is undefined) and sends (SAuth,sid,ssid) to PAi.

4.4.3. Password and Biometric Test (iii) After receiving (TestPwBio,sid,PAi,pw∗,Bio∗) from A∗, if tx(PAi)>0 then sets tested (pw) � tested(pw∗) and (Bio) � tested(Bio∗)∪PAi and tx(PAi): � tx(PAi) − 1, retrieves ⟨Reg,U,SI,pw, Bio,K⟩ and if |SI∩(tested(pw∗)∧tested(Bio∗) ∪CorrSrv)|≥t + 1andifpw∗� pwandBio �Bio∗, then returns sk to A∗and marks the record compromisedandresponsestoA∗with“correctguess”, else returns FAIL.

4.4.4. Authentication for Service Provider (i) GetCredList: Every participant can obtain all data in the public ledger of the trusted party via submitting a “retrieve” request to FDAMFA. SP then retrieves the intendedcredential PCi issuedby Nymo u from FTOPRF and accepts functionality’s assertion only if PCi⊂PC �→ . (ii) Key generation: Upon receiving (UAuth,sid,ssid, Pi,SR,sk), for |SR| � t + 1 from A∗, if there is a record ⟨Auth,P,sid,ssid,SI,SR,pw,Bio,pw′, Bio′⟩, where P∈U,SP { } then do: (a) If this record is compromised so that pw∗� pw and Bio∗� Bio or (SR⊆CorrSrv), then output (sid,sk) to player Pi. (b) Else,ifthisrecordisfresh,andifthereisarecord (P,pw′,Bio′,sk′)withpw′� pwandBio′� Bio, then sends sk′(a random key) to player Pi. (c) In any other case, picks a random key sk and sends (sid,sk) to Pi. Definition 2 (Secure DAMFA). LetΠbe a probabilistic polynomialtimeprotocolfortheDAMFAfunctionality.We say thatΠis secure if for every PPTreal world adversary A attacking DAMFA, there exists a PPTideal world simulator S such that for both the real and ideal world interactions, outputs of registration and authentication phases are computationally indistinguishable: RealA(1λ)≈IdealS(1λ). 4.5. Our Construction 4.5.1. Setup Phases We select a bilinear pairing e: G1 × G2⟶GT that is efficiently computable, nondegenerate, and three groups with prime order p. We let g1 and g2 be generators of G1 and G2, respectively, and gt � e(g1,g2) the generator of GT. Note that it is assumed to support one-way Bio-hash function H1,

which resolves the recognition error of general hash functions [62]. We consider two additional hash functions as H2: M⟶0,1 { }λ and Hg: M⟶G1.We publish params ⟵(G1,G2,g1,g2,p,hnym,H1,H 2,Hg) as the set of system parameters where hnym∈G1. The user U generates a private key K, then executes a secret sharing construction scheme on K to create secret keys for each personal agent ⟨k1,k2,...,kn⟩⟵TSS(K). U stores secret shares among personal agents. 4.5.2. Registration Phase To register a user to the system, U first chooses a passwordpwandscansherbiometricimpressionBioat the sensor. Then, U runs thefollowing stepsto register herself in the system. (i) A user runs TOPRF protocol [49] with agents to compute the secret value usk � FK(pw,Bio) as follows: (a) TheuserUpicksarandomnumberr∈Zp and computesA � Hg(pw,H(Bio))r andsendsthe message M1 � A { } to all PAi. (b) Upon receiving the message M1 � A { } from the user, each PAi computes bi � Aki � Hg(pw,H1(Bio∗))λi·ki·r by Lagrange interpolation coefficients and secret key ki (s.t. K � � i∈SRλi · ki). They return themessage M2 � bi � � to U. (c) After receiving all the messages bi from personal agents, U computes: C � � i∈SRbr−1 i � Hg(pw,H1(Bio)) K⟶usk � h(pw,C). (ii) In order to obtain a membership credential from IDP,weusePSsignaturesprotocol[47]toderivea signature on a hidden committed message as follows: (a) KeyGen(pp): The IDP runs this algorithm to generate private and public keys. This algorithm selects (x,y,y1)⟵Zp, computes (X,Y,Y1)⟶(gx 1,gy 1,gy1 1) and (X′,Y′,Y 1 ′)⟶(gx 2,gy 2,gy1 2), and sets sk⟶(X,y,y1) and pk⟶(g1,g2,Y, X′,Y′). (b) Protocol. A user first selects a random r2⟵Zp and computes C � gr2 1 · Yusk, which is a commitment on her secret key. She then sends C to the IDP. They both run a proof of knowledge of the opening of the commitment (authentication).Ifthesignerisconvinced,the IDP selects a random u⟵Zp and returns σ⟵(σ1 � gu 1,σ2 � (X· C · Ym 1)u). The user cannowunblindthesignatureσandgetavalid signature over her secret key and the message m1 by computing σ⟵(σ1,σ2/(σ1)r2) described in Sect. 3.1.

(c) Verify. To verify this signature, the user can execute this algorithm and compute: (d) Verify(pk,m,σ): e(σ1,X′· Y′usk · Y′m1 1) � e(σ2,g2). (iii) CreatePC. The user generates a protected credential with TOPRF secret key usk derived from thepasswordandthebiometric: U picksarandom number s∈Zp to generate a pseudonym as Nymo u � gs 1 · husk nym and computes an El-Gamal encryption of the credential σ with secret TOPRF values usk into a ciphertext as: PCi � [σ]usk. (iv) Proof. A NIZK proof of knowledge of the credential (PS signature [47]) works as follows: U selects random r3,t1⟵Zp and computes σ′⟵(σr3 1 ,(σ2 · σt1 1)r3). U sends σ′� (σ1 ′,σ2 ′) to theverifierandcarriesoutazero-knowledgeproof of knowledge (such as the Schnorr's interactive protocol) of m, usk, and t1 such that

π � NIZK

s,m1,t1,usk �: Nymo u � gs 1 · husk nym∧PC i � Encusk(σ)∧e σ1 ′,Y � usk · e σ1 ′,g2 � t1 · σ1 ′,Y1 � m1 � e σ2 ′,g2 � e σ1 ′,X �

⎛ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎜ ⎝ ⎫ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎬ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎪ ⎭ ⎞ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎟ ⎠. (7)(v) At the end of this phase, U submits the resulting values (PCi,π,Nymo u) to the public ledger nodes whereπ isaproofofknowledgeontheNymo u and the PCi. If the signature verifies successfully, output 1, otherwise 0. The nodes should accept values to the ledger if this algorithm returns 1.

4.5.3. Authentication Phase In this phase, a user authenticates herself to the serviceproviderandestablishesasessionkeywiththe service provider. The following steps are executed by U, PA, and SP: (i) First of all, the server chooses a secret key y⟵Zp and computes Z⟵gy 1. Then, SP generates a signature σs on message Z (i.e., Schnorr'ssignature[55])usingitssecretkeyand sends the message M1 � Z,σs to the user. (ii) When receiving a pair (Z,σs), the client verifies whether σs is valid on message Z under the SP's publickey.Ifσs isvalid, U insertspw∗andscans her personal biometric impression Bio∗at the sensor. (iii) Theuserinteractswithpersonalagentsandruns the necessary steps to compute the TOPRF protocol

FK(Bio∗,pw∗) � usk � h(pw∗,i∈SRbr−1 i). Then, U decrypts ciphertext [σ]usk with the TOPRF secret key usk to recover the credential σ. (iv) Show: The user creates a NIZK π ensuring that the credential is well-formed and the credential related to the same secret values as her pseudonym. Here we prove: (1) she knows a credential on the ledger from the IDP, (2) the credential includes the secret key as her pseudonym, (3) she possesses of a credential (signature). We use the bilinear maps accumulator [57] to accumulate the group elements g1,...,gn instead of, e.g., the integers 1,...,n { }. In addition, Camenisch et al. [57] describe an efficient zero-knowledge proof of knowledge such as Schnorr's protocol [55, 56] thatacommittedvalueisinanaccumulator.See Refs. [57, 63] to find how this proof works. U runs the following steps to authenticate herself: (a) The user selects a random number r4∈Zp to generate a pseudonym Nymv u � gr4 1 · husk nym for communication with service providers. (b) U picks random numbers d,t2⟵Zp and computes a randomized commitment credential (like in the previous step) as σ′⟵(σr2 1 ,(σ2· σt2 1)r2). (c) Then, U calculates D � gd 1, a secret session key SK � Zd � gy·d 1 and Hmac(SK,D,Z). (d) For a set of credentials PC ��→ , U computes an accumulator and witness as AC � Accumulate (params,PC ��→) and ω � GenWitness(params, PC ��→ ,PC∗ i), carries out a zero-knowledge proof of knowledge of the credential, and outputs the following proof of knowledge π such that

NIZK

usk,ω,d,t2,m,r4 : AccVerify(params,AC,ω) � 1∧ e σ1 ′,Y usk · e σ1 ′,g2 t2 · σ1 ′,Y m � e σ2 ′,g2 e σ1 ′,X ∧ PCi � Encusk(σ)∧D � gd 1∧Nym v u � gr4 1 · husk nym

(8)

Finally, U sends the message M4 � Nymv u,D, Hmac,π}, to the service provider. (i) After receiving the message M4 � Nymv u,D, Hmac, π}, from the user, the service provider first scans throughtheledgertoobtainasetPC ��→ consistingofall credentials belonging to IDP. First, SP computes the accumulator AC � Accumulate(params,PC ��→). Then,

it verifies that π � 1 is the aforementioned proof of knowledgeonPCi andNymv u usingtheknownpublic values. If the proof verifies successfully, output 1, SP computesthesessionkeyasfollows:SK � Dy � gy·d 1 . Then, SP computes Hmac∗(SK,D,Z) and checksHmac � Hmac∗.Ifπ � 1andHmacholds, SP acceptsSKas the session key and also the user is authentic. Notethatwecansimplysendσ′alongsidethemessageof the proof of knowledge. With this, we can prove the construction is aΣ-protocol (see Ref. [47] to see how proof of knowledge of PS signature works). 4.6. Optimization. To exploit the accumulator AC in our construction which can be computed incrementally, we consider that any node mining a new block can add this block'saccumulatortothepreviousone.Thenodestoresthe result as a new accumulator value in the transaction at the beginning of the new block, namely, the accumulator checkpoint. Peer nodes validate this computation before accepting the new block into the blockchain. With this optimization, SP no longer needs to compute the accumulator AC. Instead, SP can merely reference the current block's accumulator checkpoint and compute the secret key SKstartingfromthecheckpointprecedinghermint(instead of starting at the beginning).

Theorem 1. Our proposed protocol is secure against any nonuniform PPT adversary corrupting $t - 1$ many personal agents PA by assuming that the El-Gamal encryption, zeroknowledge proof of signature, and the TOPRF protocol are secure and also the hash function is collision resistant.

4.7. Security Proofs of Theorem 1 4.7.1. Proof Sketch. Our construction DAMFA is modular and relies directly on the TOPRF and the zero-knowledge proof. The security is then straightforwardly inherited from those algorithms: Thecredentialsecurityrequiresthatnoadversaryisable topresentacredential(guesspasswordsandbiometrics)and generate a session key, which they have not had any access to.IfweuseaTOPRFonpasswordsandbiometricofusers, thenthesecuritypropertiesofTOPRFwouldmakeithardto guess. The proof is once again twofold: (i) First, the authentication is done through a zeroknowledgeproof.Atthisstep,theadversarypresents an invalid credential or manages to build a valid proof.Hence,theadversarybreaks thesoundnessof the underlying proof of knowledge we used, or else uses a valid credential. (ii) At this step, we now assume the adversary wins by using a valid credential. We now rely on the obliviousness of the TOPRF. We interact with a TOPRF challenge to answer every adversarial request, and at the end, we can use the (valid) credential output by theadversary tobreakthe TOPRF obliviousness, which leads to the conclusion.

4.7.2. Anonymity. During the registration phase, when a user reveals her pseudonym but does not (intentionally) reveal her secret key usk, no adversary should learn any information about the secret key or the identity. Besides, duringtheauthenticationphase,auserproveshercredential using zero-knowledge proof, which reveals no additional information about her secret key and identity to the SP. The simulator S is essentially an ideal world adversary that interacts with the functionality FDAMFA and the environment ξ. We also assume that our zero-knowledge signature of knowledge includes an efficient extractor and a simulator and also that the signature is unforgeable. To guarantee that the view of the environment in the ideal world is indistinguishable from its view in the real world, it has to invoke the real-world adversary A by simulating all otherentitiesfor A'.Then,forthemostparts,thesimulator follows the action of adversary A'appropriately. 4.7.3. Description of the Simulator. Once the adversary registers a new user to the system via storing a tuple (Nymo u,PCi,πi) to the bulletin board, the simulator registers thisuserintheidealworldviathefollowingprocess.Itmakes aninterfacebetweenhonestpartiesintherealworld(whichare theuserUandn − t + 1personalagentsdenotedbyPAi where i � t,...,n wlog. since all personal agents in our solution are identical) and corrupted parties in the ideal world (which are theserviceprovider SP and t personalagentsdenotedby PAic where ic � (1,...,t). The simulator behaves as follows:

(1) Registration (1) Upon receiving (Reg,sid,U,SI) from FDAMFA, ignores it if |SI|≠PAn. Otherwise, records ⟨U,sid,SI⟩ and sends (Send,(sid,0),U,PA,SI) to A'for all PA∈SI. If FTOPRF sends (K,pw,Bio), records it. Remark 1. Since S simulates PAic in the ideal world, S receives whatever they receive from FDAMFA. (2) After receiving (sid,PAic,PCi,Nymo u,πi) from A'forsome PAi∈SI, it checks if it has a record of(U,kic,Nymo u)onitslistofusers.IftheuserwithNymo u exists, then S retrieves K associated

with (U,kic,Nymo u) and proceeds. The simulator then employs the knowledge extractor to obtain usk. If it is notonthelist, S followstheprotocoltoregisterNymo u as a user by choosing a random password pw∗and Bio∗. It generates secret shares kic ′ on K for eachcorruptedpersonalagent,records ⟨Reg,U,sid,SI,pw∗,Bio ∗,kic,K⟩, and sends ⟨kic⟩ to PAic∈SI and A′. (3) Upon receiving (RegComplete,sid,SI) from A′, retrieves ⟨Reg,U,sid,SI,pw∗,Bio∗,kic,K⟩ computes a pseudonym Nymv u and a credential PCi ′� h · gusk whereusk ic � FK(pw∗,Bio∗). It records ⟨Nymv u,PCi ′,U,SI,uskic⟩and sends (sid,PAic,PCi,Nymv u,πi) to its public ledger and A′ where πi is proof of knowledge. S stores

(pw∗,Bio∗,K,uskic,Nymv u,PCi ′,πi) in its list of grantedcredentials. Remark 2. When an honest user wants to establish a credential through the functionality, the simulator creates a credential and uses the extractor of the signature of knowledge to simulate the associated proof. It then transmits the credential information (PCi ′,πi,Nymv u) to the trusted store. (2) Authentication (1) Upon receiving (Auth,U∗,sid,ssid,SR) where |SR| ≥t + 1 from A′, retrieves ⟨Nymv u,PCi ′,U,SI,uskic⟩corresponding to U as stored in the registration phase. If there is a set (Bio,pw,K) stored in the registrationphaseand uskic isdefined,thenexecutes the TOPRF protocol with each personal agent using the password pw∗and Bio∗and receives ρic � T(p, (pw∗,Bio∗)) from FTOPRF andsends (Auth,sid,ssid, U,SR) to A′. Remark 3. The initialization also specifies a parameter p used to identify a table T(p,.) of random values that define the proper PRF values computed by the user when interactingwithanysubsetof t + 1honestserversfromthesetSI. An additional parameter p∗, and corresponding tables T(p∗,.),canbespecifiedbytheadversarytorepresentrogue tables with values computed by the user in the interaction with corrupted servers (see more on this [49]). (2) Upon receiving (Auth,sid,ssid,U,ρic) from FTOPRF, SrecoversSRanduskic correspondingtoUasstored duringtheregistrationphaseinthedatabase(ignores this message if no corresponding tuples exist). S checks ρic � uskic and if each PAic used the correct corresponding shareic � (uskic,kic) values. Ignores this message if either of the following conditions fails: if ρic � uskic then |S|tx(p,S)>0|>t or all servers in SR are honest. Otherwise, sends (Auth,sid,SR,pw∗,Bio∗,sk) to FDA MFA where sk is a random secret key and sets for (flag,pw∗,Bio∗,sk) as follows: (a) Case 1: Correct shareic � (ρic,kic) employed by the adversary in the real protocol. S detects this by verifying that uskic � ρic. Therefore, S sets (flag,pw∗,Bio∗,sk) � (1,.,.) and sends (uskic,kic) in its database to FDAMFA where uskic,kic was sent by FDAMFA. (b) Case 2: Otherwise, incorrect uskic,kic employed by the adversary in the real protocol. S detects this by verifying that uskic≠ρic. So, S sets (flag,pw∗,Bio∗,sk) � (0,.,.) and defines x as the set of values pw and Bio in the dictionary suchthatT(p∗,(pw,Bio))isdefined.Foreveryx in lexicographic order, sets v: � T(p∗,x) and checks if v � uskic. If so, sets (flag,pw∗,Bio∗,sk): � (2,x,sk∗) and breaks the loop. If the above loop processes all pw and

Bio without breaking, sets (flag,pw∗,Bio∗,sk) � (0,.,.). (3) On receiving (Auth,sid,ssid,SR,x � pw∗,Bio∗ ⍰ ⍰) from party P∈(U,A′) and (Auth,sid,ssid,P,ρic) fromA′,recoversuskic correspondingtoUasstored in step 1. It ignores this message if either of the following conditions fails: If ρic � uskic then |S|tx(p,S)>0|>t or if all servers in SR are honest. Otherwise, picks T(p∗,x)⟵0,1 { }l if it has not been defined and sends (Auth,sid,ssid,T(p∗,x)) to A′. If ρic � uskic (without resulting in the failure of conditions) then adds every PA∈SR to tested(x) and sends (TestPwBio,sid,PA,pw∗,Bio∗) to FDAMFA. If FDAMFA replies sk, then records it.

Remark 4. FDAMFA employs the ideal user-provided password and biometric test in the ideal world. Therefore, if the adversarial personal agents in the real world acted honestly, itmeansthatthesimulatorprovidedcorrectpairs (uski,ki). Then, the calculated credential and pseudonymous will be valid(consistingintheledger)sinceitiscomputedusingthe actual password and biometric. On the other hand, if personal agents acted maliciously in the real world, S would have detected this in the previous step and would have provided wrong pairs to FDAMFA in the ideal world. So, in both worlds, the response will be invalid. (4) Upon receiving (Auth,sid,ssid,SR,Nymv u,PCi) from FDAMFA, Sforwards ⟨Nymv u,PCi⟩ totheA'inthe real world. (3) The Indistinguishability (i) GameReal. This is the real world: the system constructedinthisworkisrunbetweenn − t + 1honest parties and t parties controlled by the adversary. (ii) Game1. This is identical to GameReal except that the encryption generated in the registration phase by honest users is replaced with a simulated one. Indistinguishability between GameReal and Game1 comes from the El-Gamal encryption security properties. (iii) Game2. This is identical to Game1 except that in TOPRF,eachshare(bi andusk)generatedbyhonest users using an actual password pw and biometric Bio is replaced by pw∗and Bio∗chosen randomly. Since, S does not have the correct password and biometric, indistinguishability between Game1 and Game2 comes from the indistinguishability of the TOPRF algorithm and TSS construction. (a) Reduction 1. The TOPRF security ensures that senders (adversarial personal agents) cannot distinguish between the receiver (the simulated user) input, whether they are the actual password pw and Bio or another randomly chosen pair of password pw∗and biometric Bio∗. (b) Reduction 2. The TSS security ensures that less than the threshold number of agents cannot

reconstruct the secret and also cannot check if theshares are indeedrelatedtothesamesecret. Therefore, there is no efficient way for the adversary to distinguish this from real behavior since one more agent needs to be corrupted to mount a successful offline attack. (iv) Game3.Thisgameisidenticalto Game2 exceptthat anauthenticationresponse(Nymv o andPC∗ i),which are two random group elements generated by the adversary will be rejected if the extracted secret key does not fulfill the requirements. Indistinguishability between Game2 and Game3 comes from the verified consistency of the bilinear pairing algorithm and the simulation breaks the soundness of the underlying proof of knowledge we used before (assuming that there is no hash collision). (v) Game4. This is the world simulated by S. It is not hard to check that Gameideal is identical to Game4. We already know that the possibility of TOPRF and NIZK proofs to break is negligible.

5.Implementation Inthissection,weillustratethepracticabilityoftheproposed protocol. To this end, we provide the public ledger part which is realized by well-known blockchains, namely, Namecoin and Ethereum. The results are summarized in Table 1. Here, initial data size shows the size of the blockchainneededfordownloadingandstorage.Initialsynctime is the time required to sync and connect to the blockchain. Confirmation time is the time required to confirm that the data are uploaded in the blockchain.

5.1. Namecoin Implemention. The public ledger can be implemented by a blockchain system. One of the smooth waystorealizeapublicledgerisusingNamecoinblockchain. Namecoin allows for registering names and stores related values in the blockchain, which is a securely distributed shared database. It also enables a basic feature to query the database and to retrieve the list of existing names and associateddata.Thus,wecanstorecredentials,scanthembased on namespace, and then verify them. We execute the following steps in order to participate in the Namecoin system and store credentials by the namecoin id as pseudonyms: (i) We need to install a Namecoin client that has a full copy of the Namecoin blockchain and keep it in sync with the P2P network by fetching and validating new blocks from connected peers. We use implementationoftheNamecoinclient[64],which can be controlled by HTTP JSON-RPC, command line, or graphical interface. It spontaneously connects to the Namecoin network and downloads the blockchain. (ii) The Namecoin client also creates the user's wallet, whichincludestheprivatekeyofNamecoinaddress of the user.

(iii) Tosavecredentialsintheblockchain,theuserneeds to register a namespace “id/name” as the owner of the name by paying a very small fee (currently 0.0064 USD). An id name can be registered using the Namecoin graphical interface or commands “name_new” and “name_firstupdate.” The following description shows how the id name in Namecoin namespace is registered and how those names can be accessed. namecoind name-new id/3608a30756b0... The output will look like this: [“0e0e03510b0b0b7dbba6e301e519693f6. 8062121b29f3cd3a6652c238360d0d0”, “9f213ff4a582fd65”]. This transaction shows a hashed version of the name, salted with a random value (which is “9f213...” for transaction ID “0e0e0351...”). (iv) The user can store arbitrary data as descriptions (whichcontainsacredential)forNamecoinkeysusing theJSONformat:thefollowingcodescanbeasimple example of the JSON value of an identity name: namecoind name_firstupdate id/3608... Output: {“description”: “28790de641755e77d1. 3382229156f5c26a9dd8a9673006b...”, “namecoin”: “NBvmSUQbRGu...”} Subsequently, the update has been confirmed and transactions have been added to the blockchain. The user has a fully valid credential. To show the credential, SP scans through the list of added names and retrieves all credentials via a graphical interface or commands like the following code: namecoind name_list Output: [{“name”: “id/3608a30756b07e...”, “value”: “28790de641755e77d13382. 229156f5c26a9dd8a9673006b15...”, “address”: “NBvmSUQbRGunCS...”, “expires_in”: 36000}]. (1) Cost: Initially, a reasonable transaction fee of either 0.00 or 0.01 NMC is charged. We can choose this fee basedonhowfastwewanttoprocessatransaction. (2) Latency: Namecoin and Bitcoin both attempt to generate blocks every 10minutes; on average, it takes nearly 5 minutes to see the data appear on the blockchain. In

practice,itthentakesthenecessarytimetosolidifythe transactions and the data to be verified. For Namecoin,ittakesabout2hourstoconfirmthatthedataare uploadedintheblockchain(12confirmations).Thatis why name_firstupdate will only be accepted after a mandatory waiting period of 12 additional blocks.

Remark5. Notethatthesecostsanddelaysoccur onlyonce during the setup and registration phases. They do not affect theauthenticationphase.Thus,wefocusonthecomputation time of the authentication phase that is frequently used in the authentication system (see Section 5.3).

5.2.Ethereum. Ethereum allowsustotestourdecentralized application on a local blockchain; we use a test network called Rinkeby to build our decentralized application. We can connect to the Ethereum blockchain and even perform operations such as mine blocks, send transactions, and deploy smart contracts by running an Ethereum node. (i) We run the Ethereum wallet (minst or geth command line) in order to access to Ethereum protocol and deploy our smart contract. (ii) Tostart,weneedtosinktheRinkebynetworklocally and download blockchain which takes a few hours. (iii) Create an account: Enter a password for your Rinkeby Account by geth command line or. Ethereum graphic (Minst). Geth Version: 1.8.1-stable. creates an account using geth command: geth account new. (iv) Next,obtainsomeEthersothattransactionscanbe sent. Since we used the Rinkeby testnet, their Ether can be obtained for free at the faucet website. Ether is used to pay transaction fees. (v) We can deploy smart contracts to store our credentials and names into them. For this purpose, we writeourfirstsmartcontractinSolidity(Solidityisa high-level contract language that is planned to target the Ethereum Virtual Machine (EVM)) and deploy it through Mist. A simple example code is pragma solidity 0.4.2

contract Test { string public NYM; string public Z;

function Test(string $−NYM$, string $−Z$) { v1� $−NYM$; v2� $−Z$; } }

Table 1: Comparison of public ledger instantiations. Properties Namecoin Ethereum (Rinkeby) Initial data size ≈5.08GB ≈5.3GB Initial sync time ≈3h ≈3h Cost 0.069 USD 0.0225 USD Confirmation time 10min/2h a few seconds/3min

(vi) We can also see the option to watch previously deployed contracts and tokens. We can click on “Watch Contracts” at the bottom and enter the contract’s name and contract address. (1) Cost: All transactions need some amount of gas to motivateprocessing.Atransactionfeeisbetween0and 0.000424ETHERdependingonhowfastwewantto approve the blockchain transaction. (2) Latency: Ethereum creates a new block every few seconds so thatthedatawillappearontheblockchaininstantly. As mentioned in Ethereum Blog, 10 confirmations are sufficient to achieve a similar security degree as that of 6 confirmations in Bitcoin. It takes around 3minutestoconfirmthetransaction/data.Notethat these costs and delays occur only once during the setup and registration phases.

5.3. Performance of the Authentication System. We now examine the performance of our anonymous authentication system.Therearetwomainsteps:theregistrationphaseandthe authentication phase. However, since time-critical operations inbothregistrationandauthenticationphasesarethesame,we concentrateourevaluationontheefficiencyoftheseprocesses. These processes include OPRF, issuing/receiving a credential, and proving knowledge of the signature and pseudonym. To simplify the evaluation criteria of the experiment results, we only assume a simple policy with a threshold t � 2 for two agents.TheexperimentisbasedonalaptopwithIntelCorei56200UCPU2.30GHz,8.00GBRAM,and64-bitUbuntuOSin Java 8, building upon the upb.crypto library (available at https://github.com/cryptimeleon) [65]. This library offers elliptic curve math and several useful building blocks for anonymouscredentialslikeSanderssignatures[47],Pedersen’s commitment [58], Nguyen’s accumulator [66], Shamir secret sharing, generalized Schnorr protocols, proofs of partial knowledge [67], Damg˚ ard’s technique for concurrently blackboxsecureSigmaprotocols,andtheFiat-Shamirheuristic[56]. Table2showsthecomputationalperformancesoftheprotocols over50iterations.Forissuingandprovingprotocolsinsucha waythatacertainpolicyissatisfiedbyacredential,weassume equality of two attributes as Policy: StuID�“11111” and GENDER�“male” and credential: certifying only these attributes.

5.4. Computational and Communication Complexity. We analyze the communication and the computation complexity of our proposed protocol using the size of each element exchange involved in our protocol, the number of exponentiation needed for issuing a credential (executed only once in the registration phase) and the proving of a credential (the most frequently executed phase),

respectively. We show the following efficiency analysis in Table3. r, t, EG1,and P denotethenumberofattributesthat can be certified, the number of agents that need to be connected,thecostofexponentiationinG1,andthecostofa pairing computation, respectively. By POK EG1[n] ⍰ ⍰ (resp. POK P[n] { }), we denote the cost of proving knowledge of n secrets involved in a multi-exponentiation (resp. pairingproduct) equation, and Ver(POK) indicates the cost of verifying this proof.

5.5. Comparison. We provide a comparison of DAMFA with some of the most popular SSO schemes in Table 4. WecompareDAMFAwiththeaboveschemesintermsof Decentralization (Decent.), Passive verification (PV), Multi-Factor(MF),Formaldefinitions(FD),Anonymity (Anony.), and Selective

Disclosure (SD) attributes. Decent denotes the decentralization of the SSO schemes (i.e.,userauthenticationprocessnolongerdependsona trusted third party). We provide this by applying a distributed transaction ledger and the blind issuing protocol. PV shows that service providers can verify users (who have registered a particular credential) without requiring interaction with an identity provider. We fulfill this property using a distributed transaction ledger and anonymous credentials. Anonymity guaranteesthatnoonecantraceorlearninformationaboutthe user's identity during the authentication process. We fulfill this property by applying NIZNP+SP signature+ Pseudonym. Here,●denotes that it is unfeasible for IDP's to track users' sign-on activity onto different SPs. Also, it shows that it is impossible to correlate multiple accounts created from the same credential on different SPs. Subsequently,◐indicates that either IDP's s or SPs cancreateacorrelationbetweendifferentaccountsofthe sameuser.FDdemonstratesifproposedschemesprovide a formal security definition. In this case, DAMFA is the onlyschemethatprovidesaformalsecuritydefinitionand proof. SD allows to disclose a subset of user attributes and provesstatementsabouttheirattributes.Finally,toprotectthe user's private information against offline (OA) attacks, we use the TOPRF primitive. Here,◐means that other related schemesareresistantagainstofflineattacksaslongasIDPdoes notcompromiseorthetheft/loss/corruptionofauser'sdevice does not happen when they use this device as 2FA token.● means that resistance to offline attacks is satisfied even in the presence of a corrupted IDP or user's device.

Table 2: Performance of the authentication protocol. Sub-protocol Duration (ms) OPRF 30 ProveNym 6 IssCred 25 ProveCred 33

6.Conclusion In this paper, we proposed a decentralized authentication and key exchange system DAMFA (SSO scheme) under TOPRFprotocolandstandardcryptographicprimitives.The proposed scheme builds upon a trustworthy global appendonly ledger that does not rely on a trusted server. DAMFA fulfills the following properties: (1) Decentralization property means that the process of user authentication no longer depends on a trusted party. To realize such a distributed ledger, we propose using the blockchain system already in realworld use with the cryptographic currency Bitcoin. (2) Passiveverificationmeansthatserviceproviderswho have access to the shared ledger can verify users without requiring interaction with an identity provider. (3) Single sign-on property ensures that a user logs in with a single ID into the identity provider and then gainsaccesstoanyoftheseveralrelatedsystems.So, users do not need to register with each service provider individually. (4) Anonymityguaranteesthatnoonecantraceorlearn information about the user's identity during the authenticationprocess.Finally,weevaluatedthatour protocol is efficient and practical for authentication systems. Moreover, we provided comparison of our scheme (DAMFA)withsomeofthemostprominentSSOschemes.To demonstrate a more detailed analysis of the performance of our scheme, we analyzed the communication and the computation complexity of our proposed protocol using the size

of each element's exchange involved in our protocol and the number of exponentiation, respectively. We proved our construction's security via ideal-real simulation, showing the impossibility of offline dictionary attacks. Finally, we demonstratedthatourprotocolisefficientandpracticalthrougha prototypical implementation and implemented the public ledger using Ethereum and Namecoin blockchains.

Data Availability No additional data are available.

Disclosure This paper is an extended version of the paper entitled “DAMFA: Decentralized Anonymous Multi-Factor Authentication” [23], including complete proofs, formal security models, an Ethereum implementation, a comparison with other SSO schemes, a computation and communication complexity analysis, and improved experimental results.

Conflicts of Interest The authors declare that they have no conflicts of interest.

Acknowledgments This work was supported by Johannes Kepler Open Access Publishing Fund and has been carried out within the scope of Digidow, the Christian Doppler Laboratory for Private DigitalAuthenticationinthePhysicalWorld.Ithaspartially beensupportedbytheLITSecureandCorrectSystemsLab. The authors are grateful for the financial support by the

Table 3: DAMFA computation and communication complexity.

Transaction

TOPRF IssueCred ProveCred User PAi User IDP User SP Computation 2 EG1 EG1 (r + 1)EG1 + POK EG1[r + 1] ⍰ ⍰ 2EG1 + Ver(POK) 2EG1 + POK P[r + 1] { } Ver(POK) Communication (2t)|G1| |G1| + |POK| 2|G1| + |POK|

Table 4: Comparison of single sign-on schemes. Schemes Decent. PV OA Anony. MF FD SD SAML [19] ✕ ✕ ◐ ◐ ✓ ✕ ✕ OpenID [16] ✕ ✕ ◐ ◐ ✕ ✕ ✕ PRIMA [17] ✕ ✕ ◐ ◐ ✕ ✕ ≈ IRMA [68] ✓ ≈ ◐ ● ✓ ✕ ✓ EL PASSO [45] ≈ ✓ ◐ ● ✓ ✕ ✓ NextLeap [46] ✓ ✕ ◐ ◐ ✓ ✕ ✓ DAMFA ✓ ✓ ● ● ✓ ✓ ✓ Anonymity: NextLeap relies on unlinkable credentials. However, blinded credentials should be stored at IDP, which allows IDP to perform user tracking. Also,inPRIMA,signonacrossmultipleSPscanbelinked.Otherschemesdonotsupportunlinkablecredentials.—Offlineattacks◐:therelatedschemesonly fulfilledofflineattackifIDPishonest.InIRMA,theuser’sdevice(i.e.,IRMAapp)shouldbesecuretoprovideOA andanonymity.Otherwise,anyadversary who gets these can simply impersonate the user (we addressed this open problem in IRMA). —Selective disclosure: PRIMA supports proving statements aboutattributes,particularlywhentheyaredisplayedasextraattributessignedbyIDP.— Passiveverification≈:InIRMA,SPsstillrequiretointeractwithan IRMA API server during the authentication.

AustrianFederalMinistryforDigitalandEconomicAffairs, the National Foundation for Research, Technology and Development, the Christian Doppler Research Association, 3 Banken ITGmbH, ekey biometric systems GmbH, Kepler Universit¨atsklinikum GmbH, NXP Semiconductors Austria GmbHand CoKG, ¨Osterreichische StaatsdruckereiGmbH, and the State of Upper Austria.

References [1] M. Bellare, D. Pointcheval, and P. Rogaway, “Authenticated key exchange secure against dictionary attacks,” in ProceedingsoftheInternationalConferenceonTheoryandApplication of Cryptographic Techniques (EUROCRYPT), pp. 139–155, Springer, Prague, Czech Republic, May 2000. [2] D. Cameron, “Over 560 million passwords discovered in anonymous online database (2017),” 2017, https://goo.gl/ upDqzt. [3] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, “Zipf’s law in passwords,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 11, pp. 2776–2791, 2017. [4] V. Goel and N. Perlroth, “Yahoo says 1 billion user accounts were hacked,” 2016, https://goo.gl/q4WZi9. [5] Gemalto, “Year of mega breaches & identity theft: findings from the breach level index,” 2014, https://www.sbs.ox.ac.uk/ cybersecurity-capacity/content/2014-year-mega-breachesand-identity-theft. [6] I.Paul,“LinkedInconfirmsaccountpasswordshacked,”2012, http://goo.gl/UBWuY0. [7] A. Ometov, S. Bezzateev, N. Ma¨kitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, “Multi-factor authentication: a survey,” Cryptography, vol. 2, no. 1, p. 1, 2018. [8] O.Mir,R.Mayrhofer,M.H¨olzl,andT.-B.Nguyen,“Recovery of encrypted mobile device backups from partially trusted cloud servers,” in Proceedings of the 13th International Conference on Availability, pp.1–10, Reliability and Security, 2018. [9] S.Jarecki,H.Krawczyk,M.Shirvanian,andN.Saxena,“TwofactorAuthentication withend-to-endpassword security,” in Proceedings of the International Conference on Practice and Theory of Public Key Cryptography (PKC), Springer, Rio de Janeiro, Brazil, Cham, March 2018. [10] M. A. Mustafa, A. Abidin, and E. A. Ru´a, “Frictionless authenticationsystem: security& privacy analysisand potential solutions,” 2018, https://arxiv.org/abs/1802.07231. [11] V. Rimmer, D. Preuveneers, W. Joosen, M. A. Mustafa, A. Abidin, and E. A. Ru´a, “Frictionless authentication systems: emerging trends, Research challenges and opportunities,” 2018, https://arxiv.org/abs/1802.07233. [12] D. Q. Viet, A. Yamamura, and H. Tanaka, “Anonymous password-based authenticated key exchange,” in Proceedings of the International Conference on Cryptology in India (INDOCRYPT), pp. 244–257, Springer, Bangalore, India, December 2005. [13] Y. Lindell, “Anonymous authentication,” Journal of Privacy and Confidentiality, vol. 2, no. 2, 2011. [14] J. Yang and Z. Zhang, “A new anonymous password-based authenticated key exchange protocol,” in Proceedings of the International Conference on Cryptology in India (INDOCRYPT), pp. 200–212, Springer, Bangalore, India, December 2008. [15] S. Shin, K. Kobara, and H. Imai, “Anonymous passwordauthenticated key exchange: new construction and its

extensions,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E93-A, no. 1, pp. 102–115, 2010. [16] D. Recordon and D. Reed, “Openid 2.0: a platform for usercentric identity management,” in Proceedings of the second ACM workshop on Digital identity management, pp. 11–16, New York, NY, United States, November 2006. [17] M. R. Asghar, M. Backes, and M. Simeonovski, “Prima: privacy-preserving identity and access management at internet-scale,” in Proceedings of the 2018 IEEE International Conference on Communications (ICC), pp.1–6, IEEE, Kansas City, MO, USA, May 2018. [18] D. Hardt and M. Jones, “The oauth 2.0 authorization framework: bearer token usage,” 2012, https://www.omgwiki. org/dido/doku.php?id�dido:public:ra:xapend:xapend.b_stds: tech:ietf:oauth_bearer. [19] OneLogin Inc, “SAML toolkits,” 2019, https://developers. onelogin.com/saml. [20] S. Agrawal, P. Miao, P. Mohassel, and P. Mukherjee, “Pasta: password-based threshold authentication,” in Proceedings of the ACM Conference on Computer and Communications Security (CCS), pp. 2042–2059, ACM, New York, NY, United States, October 2018. [21] G. Alp´ar, J.-H. Hoepman, and J. Siljee, “The identity crisis. security,privacyandusabilityissuesinidentitymanagement,” 2011, https://arxiv.org/abs/1101.0427. [22] D. Fett, R. Ku¨sters, and G. Schmitz, “The web SSO standard openidconnect:in-depthformalsecurityanalysisandsecurity guidelines,” in Proceedings of the Computer Security Foundations Symposium (CSF), pp.189–202, IEEE, Santa Barbara, CA, USA, August 2017. [23] O. Mir, M. Roland, and R. Mayrhofer, “Damfa: decentralized anonymousmulti-factorauthentication,”inProceedingsofthe 2ndACMInternationalSymposiumonBlockchainandSecure Critical Infrastructure, ser. BSCI ’20, pp. 10–19, New York, NY, United States, October 2020. [24] S. M. Bellovin and M. Merritt, “Encrypted key exchange: password-based protocols SecureAgainst dictionary attacks,” inProceedingsoftheIEEESymposiumonSecurityandPrivacy (S&P), IEEE, Oakland, CA, USA, May 1992. [25] Augmented Encrypted Key Exchange, “A password-based protocol secure against dictionary attacks and password file compromise,” in Proceedings of the ACM Conference on Computer and Communications Security (CCS), pp. 244–250, ACM, New York, NY, United States, October 1993. [26] V. Boyko, P. MacKenzie, and S. Patel, “Provably secure password-authenticated key exchange using diffie-hellman," in Proceedings of the International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT), pp. 156–171, Springer, Bruges, Belgium, May 2000. [27] O. Goldreich and Y. Lindell, "Session-key generation using human passwords only," Journal of Cryptology, vol.19, no. 3, pp. 241–340, 2006. [28] R.GennaroandY.Lindell,"Aframeworkforpassword-based authenticated key exchange," Lecture Notes in Computer Science, Springer, in Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt), pp. 524–543, May 2003. [29] A. Groce and J. Katz, "A new framework for efficient password-basedauthenticatedkeyexchange,"inProceedingsofthe ACM Conference on Computer and Communications Security (CCS), pp. 516–525, ACM, New York, NY, United States, October 2010.

[30] F. Benhamouda and D. Pointcheval, "Verifier-based password-authenticated key exchange: new models and constructions,"IACRCryptologyePrintArchive,vol.2013,p.833, 2013. [31] M. Shirvanian, S. Jarecki, N. Saxena, and N. Nathan, "Twofactor authentication resilient to server compromise using mix-bandwidth devices," in Proceedings of the Network and Distributed System Security Symposium, (NDSS), San Diego, CA, USA, February 2014. [32] J.Bringer,H.Chabanne,andR.Lescuyer,"Software-onlytwofactor authentication secure against active servers," in ProceedingsoftheInternationalConferenceonCryptologyinAfrica (AFRICACRYPT), pp. 285–303, Springer, Fes, Morocco, April 2016. [33] O. Mir, T. van der Weide, and C.-C. Lee, "A secure user anonymity and authentication scheme using avispa for telecare medical information systems," Journal of Medical Systems, vol. 39, no. 9, p. 89, 2015. [34] S. Jarecki, H. Krawczyk, M. Shirvanian, and N. Saxena, "Device-enhanced password protocols with optimal onlineoffline protection," in Proceedings of the ACM on Asia ConferenceonComputerandCommunicationsSecurity(ASIA CCS), pp. 177–188, ACM, New York, NY, United States, October 2016. [35] D.HeandD.Wang,"Robustbiometrics-basedauthentication scheme for multiserver environment," IEEE Systems Journal, vol. 9, no. 3, pp. 816–823, 2015. [36] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," IEEE Transactions on Dependable and Secure Computing, vol. 11, no. 6, pp. 568–581, 2014. [37] D. Pointcheval and S. Zimmer, "Multi-factor Authenticated key exchange," in Proceedings of the 6th International Conference on Applied Cryptography and Network Security (ACNS),pp.277–295,Springer,NewYork,NY,UnitedStates, June 2008. [38] O. Mir and M. Nikooghadam, "A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services," Wireless Personal Communications, vol. 83, no. 4, pp. 2439–2461, 2015. [39] N. Fleischhacker, M. Manulis, and A. Azodi, "A modular frameworkformulti-factorauthenticationandkeyexchange," in Proceedings of the International Conference on Research in Security Standardisation, pp. 190–214, Springer, London, United Kingdom, December 2014. [40] R. Zhang, Y. Xiao, S. Sun, and H. Ma, "Efficient multi-factor authenticated key exchange scheme for mobile communications," IEEE Transactions on Dependable and Secure Computing, vol. 1, 2017. [41] D.Wang,Z.Zhang,P.Wang,J.Yan,andX.Huang,"Targeted online password guessing: an underestimated threat," in Proceedingsofthe2016ACMSIGSACconferenceoncomputer andcommunicationssecurity,pp.1242–1254,ViennaAustria, October 2016. [42] D.Wang,D.He,P.Wang,andC.-H.Chu,"Anonymoustwofactor authentication in distributed systems: certain goals are beyond attainment," IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 4, pp. 428–442, 2014. [43] D. Debiao He, N. Kumar, J.-H. Jong-Hyouk Lee, and R. S. Sherratt, "Enhanced three-factor security protocol for consumer usb mass storage devices," IEEE Transactions on Consumer Electronics, vol. 60, no. 1, pp. 30–37, 2014. [44] Z. Zhang, K. Yang, X. Hu, and Y. Wang, "Practical anonymous password authentication and TLS with anonymous

client authentication," in Proceedings of the ACM Conference on Computer and Communications Security (CCS), pp. 1179–1191, ACM, Vienna Austria, October 2016. [45] Z. Zhang, M. Kr´ol, A. Sonnino, L. Zhang, and E. Rivi`ere, "El passo: privacy-preserving, asynchronous single sign-on," 2020, https://arxiv.org/abs/2002.10289. [46] H.Halpin,"Nextleap:decentralizingidentitywithprivacyfor secure messaging," in Proceedings of the 12th International Conference on Availability, pp.1–10, Reliability and Security, Inria de Paris, December 2017. [47] D. Pointcheval and O. Sanders, "Short randomizable signatures," in Proceedings of the Cryptographers' Track at the RSA Conference, pp. 111–126, Springer, San Francisco, CA, USA, March 2016. [48] S. Jarecki and X. Liu, "Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection," in Proceedings of the Conference onTheoryofCryptography(TCC),pp.577–594,Springer,San Francisco, CA, USA, March 2009. [49] S.Jarecki,A.Kiayias,H.Krawczyk,andJ.Xu,"TOPPSS:costminimal password-protected secret sharing based on threshold OPRF," in Proceedings of the International Conference on Applied Cryptography and Network Security, pp. 39–58, Springer, Singapore, June 2017. [50] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612-613, 1979. [51] E. B. Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in Proceedings of the IEEE Symposium on Security and Privacy (S&P), pp. 459–474, IEEE, Berkeley, CA, USA, May 2014. [52] C. Garman, M. Green, and I. Miers, "Decentralized anonymous credentials," in Proceedings of the 2014 Network and Distributed System Security Symposium, Internet Society, San Diego, California,, February 2014. [53] C. Fromknecht, D. Velicanu, and S. Yakoubov, A Decentralized Public Key Infrastructure with Identity Retention, IACR Cryptology ePrint Archive, 2014, https://eprint.iacr. org/2014/803.pdf. [54] R. Yang, M. H. Au, Q. Xu, and Z. Yu, "Decentralized blacklistable anonymous credentials with reputation," Computers & Security, vol. 85, pp. 353–371, 2019. [55] C.-P. Schnorr, "Efficient identification and signatures for smart cards," in Proceedings of the Annual International Cryptology Conferenceon Advancesin Cryptology(CRYPTO), pp. 239–252, Springer, Aarhus, Denmark, May 1990. [56] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," in Proceedings of the Conference on the Theory and Application of Cryptographic Techniques (Eurocrypt), pp.186–194, Springer, Santa Barbara, CA, USA, March 1986. [57] J. Camenisch, M. Kohlweiss, and C. Soriente, "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," in Proceedings of the International WorkshoponPublicKeyCryptography,pp.481–500,Springer, Irvine, CA, USA, March 2009. [58] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in Proceedings of the Annual International Cryptology Conference (Crypto), pp. 129–140, Springer, Santa Barbara, CA, USA, August 1991. [59] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributedkeygenerationfordiscrete-logbasedcryptosystems," in ProceedingsoftheInternationalConferenceontheTheoryand Applications of Cryptographic Techniques (Eurocrypt), pp. 295– 310, Springer, Prague, Czech Republic, May 1999.

[60] R. Canetti, S. Halevi, J. Katz, Y. Lindell, and P. MacKenzie, "Universally composable password-based key exchange," in ProceedingsoftheAnnualInternationalConferenceonTheory andApplicationsofCryptographicTechniques(EUROCRYPT), pp. 404–421, Springer, Aarhus, Denmark, May 2005. [61] J.Camenisch,M.Drijvers,andB.Tackmann,"Multi-protocol uc and its use for building modular and efficient protocols," IACR Cryptol. ePrint Arch, vol. 2019, p. 65, 2019. [62] A.T.B.Jin,D.N.C.Ling,andA.Goh,"Biohashing:twofactor authentication featuring fingerprint data and tokenised random number," Pattern Recognition, vol. 37, no. 11, pp. 2245–2255, Nov 2004. [63] M. Chase, S. Meiklejohn, and G. Zaverucha, "Algebraic macs and keyed-verification anonymous credentials," in Proceedings of the ACM SIGSAC Conference on Computer and Communications

Security (CCS), pp. 1205–1216, ACM, Scottsdale Arizona USA, November 2014. [64] N. team, Namecoind, Sourcecode of the Namecoin-Client Reference Implementation, Namecoin, Aug 2016, https:// github.com/namecoin/namecoin. [65] K. Bemmann, J. Bl¨omer, J. Bobolz et al., “Fully-featured anonymous credentials with reputation system,” in Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES 2018), no. 42, pp. 1–10, IEEE, New York, NY, United States, August 2018. [66] L. Nguyen, “Accumulators from bilinear pairings and applications,” in Lecture Notes in Computer Science, vol. 3376, pp. 275–292, Springer, 2005. [67] R. Cramer, I. Damga˚rd, and B. Schoenmakers, “Proofs of partial knowledge and simplified design of witness hiding protocols,” in Advances in Cryptology – CRYPTO ’94, ser. LNCS, vol. 839, pp. 174–187, Springer, 1994. [68] G.Alpa´r,F.vandenBroek,B.Hampiholi,B.Jacobs,W.Lueks, and S. Ringers, “Irma: practical, decentralized and privacyfriendly identity management using smartphones,” HotPETs, vol. 2017, 2017.