To Pass or Not to Pass: Privacy-Preserving Physical Access Control

Jesús García-Rodríguez, *University of Murcia*
Stephan Krenn, Daniel Slamanig, *AIT Austrian Institute of Technology*

***Abstract*—Anonymous or attribute-based credential (ABC) systems are a versatile and important cryptographic tool to achieve strong access control guarantees while simultaneously respecting the privacy of individuals. A major problem in the practical adoption of ABCs is their transferability, i.e., such credentials can easily be duplicated, shared or lent. One way to counter this problem is to tie ABCs to biometric features of the credential holder and to require biometric verification on every use. While this is certainly not a viable solution for all ABC use-cases, there are relevant and timely use-cases, such as vaccination credentials as widely deployed during the COVID-19 pandemic. In such settings, ABCs that are tied to biometrics, which we call Biometric-Bound Attribute-Based Credentials (bb-ABC), allow to implement scalable and privacy-friendly systems to control physical access to (critical) infrastructure and facilities.**

While there are some previous works on bb-ABC in the literature, the state of affairs is not satisfactory. Firstly, in existing work the problem is treated in a very abstract way when it comes to the actual type of biometrics. Thus, it does not provide concrete solutions which allow for assessing their practicality when deployed in a real-world setting. Secondly, there is no formal model which rigorously captures bb-ABC systems and their security requirements, making it hard to assess their security guarantees. With this work we overcome these limitations and provide a rigorous formalization of bb-ABC systems. Moreover, we introduce two generic constructions which offer different trade-offs between efficiency and trust assumptions, and provide benchmarks from a concrete instantiation of such a system using facial biometrics. The latter represents a contact-less biometric feature that provides acceptable accuracy and seems particularly suitable to the above use-case.

***Index Terms*—anonymous credentials, biometrics, risk-based access control**

I. Introduction

Attribute-based credentials–also known as anonymous credentials, or simply ABCs–allow *users* (or *provers*) to receive *credentials* certifying certain pieces of personal information known as *attributes* from *issuers*. Later, users can present their credentials to *verifiers* while keeping full control over the disclosed information. That is, users can decide which attributes to disclose and which attributes to keep private, while still giving the verifier formal authenticity guarantees on the revealed information. Even more, users may also be able to prove that their attributes satisfy complex policies, involving, e.g., proofs that an attribute is above a certain threshold (e.g., for age proofs) or belong to a certain set (e.g., proving that one is vaccinated against, or recovered from, a disease), without revealing any additional information than what is required by the policy. In particular, ABC systems also give high metadata privacy guarantees, by ensuring that different actions of the same user cannot be linked, except by the disclosed information.

Attribute-based credentials were already envisioned in the 1980's by Chaum [1], [2]. The most well-known schemes are Microsoft's U-Prove [3], [4] and IBM's Identity Mixer [5]–[8]. Besides those, a large variety of schemes, fulfilling different security and privacy notions, providing different performance trade-offs, and proposing different features and functionalities, have been introduced, including, e.g., [9]–[18].

a) **Transferability problem:** Despite their benefits, most existing ABC systems suffer from the drawback of *transferability of credentials*: if credentials are purely software-based, they can be duplicated, shared, lent, or sold, hindering the adoption of the technology in the real world.

Multiple approaches to overcome this issue have been proposed. For instance, Camenisch and Lysyanskaya [5] proposed *all-or-nothing sharing*, where sharing a credential once already implies the ability to use the credential in any context, i.e., taking over the user's identity. However, as also noticed, e.g., by Adams [19], while such approaches may disincentivize users to broadly share their credentials, they do *not* prevent sharing of credentials, e.g., among close friends or family members. Also, depending on the application scenario, sharing the entire credential might only have limited impact for the legitimate owner: e.g., during the COVID-19 pandemic, digital COVID certificates (so-called "Green Pass" certificates) have seen a widespread enrolment within Europe[1]. They can be used to prove that a person has recovered from, was vaccinated against, or negatively tested for, a certain disease, and users may not bother about revealing the entire credential upon (illegitimate) sharing. An alternative approach is thus to bind ABCs to tamper-proof hardware, e.g., [20], [21], to avoid duplication of credentials. However, hardware-bound credentials either require users to carry dedicated hardware with them, or require re-issuance of credentials when upgrading hardware such as mobile phones, thus limiting their flexibility and usability. Furthermore, sharing among close family members may still not be prevented as they may have access to the same physical devices.

b) **Use of biometrics:** A natural solution is therefore to bind credentials to the physical identity of users by leveraging biometrics. That is, the idea is to encode a biometric feature

This work was done while the first author was visiting AIT Austrian Institute of Technology.

[1] https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate_en

vector as part of the attributes. Upon presentation, the user then needs to prove that she is the legitimate owner of the credential by proving that she "owns" matching biometrics, in addition to what is requested by the presentation policy. Such an approach is particularly useful for privacy-preserving physical access control.

In some scenarios, such as access control to restricted areas like a sensitive work space, e.g., critical infrastructure, where identification of a user is unproblematic or even desired, the biometrics can simply be treated as a disclosed attribute: the turnstile, acting as a verifier, could measure the biometrics of the current user, compare it to the certified and disclosed attribute in cleartext, check whether the remaining presentation policy (e.g., vaccination status, access rights, etc.) is satisfied, and let the user pass if and only if this is the case.

However, the situation is different, e.g., when performing access control to public transport, restaurants or events, where identification and linkability of users is undesirable, and no biometric information should thus be given to the verifier. In order to reach meaningful security and privacy notions, the verifier now needs to be split into: (i) a semi-trusted device measuring a user's biometrics, and (ii) the untrusted access control system acting as a verifier. After measuring a user's biometrics, this device would then send necessary data to the user and/or the verifier, and the user would prove in a zero-knowledge manner that she possesses a credential matching these biometrics, thereby however considering that two measurements of the same biometric property will typically not yield identical but only nearly identical results. To justify the necessary trust in the biometric device, the amount of operations within this device should be kept as small as possible to enable audits and certifications; furthermore, to ease the real-world adoption of such a system, only minimal requirements regarding hardware and software capabilities of this device should be made.

c) **Related Work:** Different approaches for enabling privacy-preserving authentication using attributes and biometrics have been proposed in the literature. For instance, [22]–[24] proposed solutions based on a dedicated trusted device, e.g., a smart card, carried by every user. In a nutshell, the idea is that the smart card is trusted to scan fresh fingerprints upon each presentation of the credential, and then prove that the measured fingerprint indeed matches the one encoded inside the credential. However, a solution requiring dedicated hardware per user does not scale and additionally suffers from the same usability limitations as device-bound credentials. Other approaches, e.g., by Bissessar et al. [25], for binding credentials to physical identities use fuzzy extractors (FEs) [26]. On a high level, FEs take as input a sample from a noisy source (e.g., biometric data), and output the same digest as long as the two samples are sufficiently close to each other. While FEs are an attractive object [26]–[30], a major drawback for their practical use are the storage requirements (or bandwidth requirements when transmitted) of the helper data required by biometric data. This is typically in the hundreds of MB or even GB [30], which makes them unusable in the setting of this paper. Even then, the accuracy levels achieved by such constructions ($\ll 90\%$) are very far from current biometric practices [31], [32].

In another line of work, e.g., [33], [34] suggested efficient solutions based on functional encryption; however, they consider a different setting requiring preregistered (encrypted) biometrics at the service provider, and also do not consider attributes beyond biometrics.

Finally, Adams [19] proposed a solution close to ours with a focus on non-transferability: intuitively, the biometrics sensor encrypts the measured biometrics for the user, and hands a commitment to the value to the verifier. The user then computes a zero-knowledge proof of knowledge showing that the biometrics certified in the credential match those in the commitments sent by the sensor. Adams [19] discusses a system for generic biometrics based on the one-show credential approach in U-Prove [3], [4]. Moreover, a similar approach based on multi-show Camenisch-Lysyanskaya credentials [6] is described by Camenisch et al. [35].

While the work in [19], [35] presents important conceptual contributions towards what we call Biometric-Bound Attribute-Based Credentials (bb-ABCs), it leaves open a number of important questions. On the conceptual side, security is either omitted or only argued on an ad hoc basis and thus no formal treatment is available so far. More importantly, these works do not assess practical aspects when deploying such systems such as suitable biometric features, let alone a practical implementation and performance evaluation. Actually, Adams [19] concludes that his approach "is likely to be too inefficient or too complex for many practical environments". One of our aims is to show that bb-ABCs are indeed practical and can be a valuable tool in real-world applications.

Finally, we want to mention the independent and concurrent work by Hesse at al. in [36], who introduce so called anonymous credentials with visual holder authentication. This setting introduces an additional physical device, e.g., a smartcard, that is capable of displaying a picture of the holder, to be verified personally by the verifier, and to take part in the showing of an anonymous credentials. Unfortunately, the requirement for additional dedicated hardware and manual checks, makes this approach not suitable for our main application, i.e,. risk-based access control in pandemics.

A. Applications

a) **Risk-Based Access Control in Pandemics:** "Green Pass" certificates have been broadly deployed during the COVID-19 pandemics. They can be used to control physical access to (critical) infrastructure and facilities in pandemics and represent an important measure to reduce the exposure of individuals to infectious diseases, and so contribute to maintaining the continuity of operation of (critical) infrastructure. Thus, they can be seen as a measure to implement risk-based access control (RiBAC) [37][2]. Such health certificates are typically realized as documents containing some personal attributes plus information about received vaccinations as well

[2] The term "risk-based access control" is already used for access control mechanisms where access decisions are based on quantified risk estimates [38], [39]. As argued in [37], however, it also provides a good intuition for pandemic situations where the aim is to *reduce risk* via access control.

as recovery and test information, which are signed by some authority. Verification is then performed by scanning a QR-code including the data and its signature and checking the personal attributes against a physical identity document (e.g., passport). This current technology, however, comes with some drawbacks. First, the checking procedure is time consuming and does not scale. Secondly, it is not desirable from a privacy perspective as all information within this document is revealed. Here, it is important to note that revealing the status of whether vaccinated, recovered, or tested is not necessary for making an access decision: it is sufficient to know that one of those criteria is satisfied. This matters when the decisions about vaccination are delicate or controversial[3] and especially since privacy seems to be an important aspect why people prefer non-scalable paper-based certificates over digital ones [40]. Consequently, we consider bb-ABCs that i) allow to prove the status, e.g., being vaccinated, recovered or tested, together with ii) a fast and contact-less biometric feature encoded in the credential, i.e., facial biometrics, as a scalable and privacy-friendly RiBAC approach in times of a pandemic.

b) **Additional Application Domains:** The concept of bb-ABCs can also be beneficial in other application domains besides pandemics. For instance, thrift shops (also known as charity stores), mainly offer donated goods to provide affordable shopping opportunities to a less prosperous clientele. To avoid misuse of the system, customers usually need to present a certificate (e.g., a wage statement) and an identity card, thereby fully identifying themselves. Using bb-ABCs, customers could receive a credential certifying their eligibility to take advantage of the offer without the need to re-identify themselves upon every purchase, without the risk of misuse or transfer of the credential, thereby potentially reducing the perceived discrimination and increasing clients' willingness to take advantage of the offer. Similarly, bb-ABCs could be used to bind coupons, such as food stamps or ration stamps in case of major crises, to their physical owners, thus slowing down the emergence of a black market. Finally, bb-ABCs can reduce coercion of legitimate owners by rendering sharing of credentials impossible.

B. Our Contribution

In a nutshell, we provide a framework, generic constructions, concrete instantiations, and feasibility micro-benchmarking of Biometric-Bound Attribute-Based Credentials (bb-ABCs).

More precisely, our contribution can be summarized as follows:

- We provide a detailed definitional framework for bb-ABCs, considering a scenario where the verifier is equipped with a biometric device which is semi-trusted by both, users and the verifier. To the best of our knowledge, while this setting has been considered in previous works, security has always been argued on an ad hoc basis, and a rigorous formal framework is not available in the literature.
- We present—and prove secure—two generic constructions. As the framework itself, the constructions are agnostic to the concrete biometric feature being used. The first construction defers the matching of biometric templates to the reader device. The second construction leads to a more transparent system, where less trust has to be put in the reader device, as the matching is done (and proven in zero-knowledge) by the user, at the cost of a higher complexity.
- We then present instantiations of our generic constructions. While the first construction can efficiently be instantiated for any biometric feature, achieving practical efficiency in terms of computation and communication for the second construction turns out to be more challenging, as it requires to prove in zero-knowledge the actual biometric matching algorithm. Our solution is based on face recognition using cosine similarity [41], with parameters recommended for achieving 95% accuracy for "faces in the wild" [42], i.e., using real-world non-standardized images, which is most realistic setting for the scenarios considered in this paper.
- Finally, we show the practical efficiency of our constructions, by providing fine-granular micro-benchmarks of all relevant steps of our protocol using representative device profiles for user, reader and verifier. While the first construction causes virtually no overhead, the second instantiation adds a total overhead of around 2.1s to a showing of a comparable non-biometric-bound credential.

II. Preliminaries

In the following, we introduce the notation being used, as well as the necessary background required for the remainder of this paper.

A. Notation

We denote the main security parameter by λ. We use $out \leftarrow\$ A(in)$ to denote that out is the output of a randomized algorithm A on input in; similarly, we write $x \leftarrow\$ \mathcal{S}$ to denote that x was sampled uniformly at random from a set $\mathcal{S}$. We use $\boldsymbol{v}$ to denote the vector $(v_1, \ldots, v_n)$. We say that a function $\mathsf{negl}(\lambda) : \mathbb{N} \to [0, 1]$ is negligible, if it vanishes faster than any inverse polynomial.

B. Biometric Authentication

Biometric authentication uses unique biological characteristics to verify that a person is who she claims to be. Such systems are based on biometric templates, which are mathematical representations of biometric features such as fingerprints, retina scans, voice recordings, facial images, or behaviour. Biometric templates will be denoted by Bio throughout this paper. Additionally, we will use a_{Bio} to refer to a biometric template to which a credential is bound, that is, the template that is included as an attribute in the credential.

At a high level, a biometric matching algorithm is now a system that takes as inputs two templates Bio_1 and Bio_2, and outputs a measure for similarity (or matching score),

[3] https://www.aarp.org/work/careers/older-workers-vaccine-status/

based on which a decision has to be taken, e.g., whether to grant or to deny access. Note that due to the nature of biometric measurements, this decision implies false positives and/or false negatives with a probability that depends on the specific scheme, parameters and implementation being used.

The ambition of this work is to realize privacy-preserving access control which is as secure as the underlying authentication scheme. Consequently, we will write $\mathcal{M}(\cdot,\cdot)$ to denote a matching algorithm enhanced with the final decision making; that is, $\mathcal{M}(Bio_1, Bio_2) = 1$ if and only if the templates coincide for the selected matching and decision algorithms, and 0 otherwise.

C. Cryptographic Building Blocks

The generic constructions presented in the paper rely on various thoroughly studied cryptographic building blocks. Here, we give a brief overview on the main interfaces of these protocols, and provide informal descriptions of their security properties; for full details we refer to the original literature. Note that all schemes presented in the following may have setup algorithms to generate public parameters, which are omitted in this informal description for readability reasons.

a) **Digital signatures:** A digital signature scheme enables a receiver to verify the authenticity of a received message (or vectors of messages). Such a scheme consists of the following algorithms:

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\$\ \Sigma.\mathsf{KeyGen}(1^\lambda)$: Generate a key pair.
- $\sigma \leftarrow\!\$\ \Sigma.\mathsf{Sign}(\mathsf{sk}, \boldsymbol{a})$: Sign a message $\boldsymbol{a}$ using a secret key.
- $b \leftarrow \Sigma.\mathsf{Verify}(\mathsf{pk}, \boldsymbol{a}, \sigma)$: Verify a signature with respect to the public key.

Besides correctness, digital signature schemes need to satisfy *existential unforgeability under chosen messages attacks* (EUF-CMA), meaning that no adversary—not having access to the secret signing key—can come up with a valid signature on a new message, even if it was granted access to a signing oracle for arbitrary messages of its choice. For a formal definition we refer to [43].

b) **Commitment schemes:** A commitment scheme allows a party to bind itself to a specific value, without revealing it to anybody else, with the ability to later disclose the value. It consists of the following algorithms:

- $(C, V) \leftarrow\!\$\ \mathcal{C}.\mathsf{Commit}(m)$: Generate a commitment C to a value m, and opening value V.
- $b \leftarrow \mathcal{C}.\mathsf{Open}(C, V, m)$: Verify the validity of a commitment and opening.

A commitment scheme is *perfectly hiding*, if C does not contain any information about m in an information-theoretic sense. Furthermore, it is *computationally binding*, if it is computationally infeasible to generate a commitment and two accepting openings to different messages. For formal definitions, cf. Pedersen [44].

c) **Authenticated encryption:** An *authenticated encryption scheme* is a symmetric encryption which simultaneously guarantees confidentiality and authenticity of data. Such a scheme consists of the following algorithms:

- $sk \leftarrow\!\$\ \mathcal{E}.\mathsf{KeyGen}()$: Generate a secret key.
- $ae \leftarrow\!\$\ \mathcal{E}.\mathsf{Encrypt}(sk, m)$: Encrypt m under a secret key sk.
- $m \leftarrow \mathcal{E}.\mathsf{Decrypt}(sk, ae)$: Decrypt ciphertext.

An authenticated encryption scheme guarantees that no adversary having access to encryption and decryption oracles can either learn any information about plaintext except their lengths, or forge any new valid ciphertexts for which decryption will not abort, see, e.g., Barwell et al. [45].

d) **Non-interactive zero-knowledge proofs:** A non-interactive zero-knowledge proof of knowledge (NIZK) allows a prover to convince a verifier that it knows a secret piece of information (i.e., a witness w to a statement x satisfying a binary relation $\mathcal{R}$), without revealing anything beyond what is already revealed by the claim itself. NIZKs consist of the following algorithms:

- $pt \leftarrow\!\$\ \Pi.\mathsf{Prove}(x, w, ctx)$: Generate a NIZK pt bound to ctx such that $(x, w) \in \mathcal{R}$.
- $b \leftarrow \Pi.\mathsf{Verify}(pt, x, ctx)$: Verify a proof pt.

A NIZK is *zero-knowledge*, if pt does not reveal any additional information about w than what is already revealed by x, as (potentially knowing a simulation trapdoor) proofs with an indistinguishable distribution can also be generated only knowing x. On the other hand, *simulation-sound extractability* means that it is computationally hard to generate a valid proof for a statement and a context without knowing the corresponding witness, even after having seen arbitrarily many simulated proofs for statements chosen adaptively by the adversary [46].

For convenience, we use the notation introduced by Camenisch and Stadler [47], where a NIZK, bound to the context ctx, for values (α, β) that fulfil the right-hand side condition is denoted by:

$$pt \leftarrow\!\$\ \mathsf{NIZK}[(\alpha, \beta) : g^\alpha h^\beta = y_1 \wedge \alpha \leq x](ctx)$$

III. Framework for bb-ABCs

In this section, we define the syntax of Biometric-Bound Attribute-Based Credentials (bb-ABCs), and formally define the security requirements that such a system needs to satisfy.

A. Protocol Definitions

A bb-ABC system consists of four actors: issuers (I), provers/users (U), and verifiers (V) as in ABC systems, as well as a *reader device* (R) for recording a user's biometrics. In the following, we introduce how they interact through the protocol algorithms, and later provide their formal definitions and an overview on the system (cf. also Section VI for further discussion the system and its design).

Upon initialization of the system, public parameters are established using ParGen.

Issuers can then generate their key material with I.KeyGen, and issue credentials to users using the non-interactive I.IssueCred algorithm. Issuance is modeled as a non-interactive algorithm for simplicity and to not disguise the specificities of our system by particularities like biometric enrollment during credential issuing or extensions (e.g., non-frameability, pseudonyms, or revocation), which can be modularly added using standard techniques, cf., e.g., [11], [48] or Section VI-B. Having received a credential, users can verify its validity using U.VerifyCred.

With a valid credential, users can present it to a verifier, supported by a reader device. Using U.GenEph, users generate ephemeral cryptographic material for each presentation process. The reader device is in charge of measuring and processing the fresh biometric template, and deriving from it some data for the user with R.GenEphUser. The user then derives a presentation token from her credential using U.Present.

In order to validate a presentation token, the verifier may interact with the reader device, computing its input to the reader using V.InputGen, and receiving back output from R.GenEphVerifier. The verifier finally checks the presentation token using V.Verify.

Note that the R.GenEphUser and R.GenEphVerifier are defined explicitly as separate algorithms, whose output is used as input for presentation and verification, to improve the readability and expressiveness of the framework and security models. In a practical application, they could be subroutines started by the user and verifier. Similarly, all parties keep state as required by the protocol, but we do not make it explicit for notational convenience. Note that this will be short-term state during presentation, mostly for ephemeral keys. In fact, depending on the protocol instantiation, state may not involve any sensitive material— e.g., as a sneak peak into Construction 2, only a (perfectly-hiding) commitment of the fresh biometric scan is temporarily stored.

The formal specifications of the interfaces are now as follows.

a) **Setup.:** In our security model, we will assume that these parameters are honestly generated; in practice, this could be enforced, e.g., by generating them (once and for all) via an MPC ceremony.

- $pp \leftarrow\$ \mathsf{ParGen}(1^\lambda)$: Set up public parameters pp from the security parameter, e.g., specifying the number of attributes and their domains, groups, etc. These parameters will be implicitly used as input for all other algorithms.

b) **Key generation and issuance:** To obtain a credential, the following steps by the issuer and user are required.

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{I.KeyGen}(pp)$: Generate a key pair for the issuer.
- $\sigma \leftarrow\$ \mathsf{I.IssueCred}(\mathsf{sk}, a_{Bio}, \boldsymbol{a})$: Create a credential σ on attributes $\boldsymbol{a} = (a_1, \ldots, a_n)$, where a_{Bio} is the biometric attribute that binds the credential to the user and a_i are her identity attributes. Proper verification of the requested attributes (e.g., through a physical process) is out of scope.
- $b \leftarrow \mathsf{U.VerifyCred}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a})$: Verify the validity of an issued credential.

c) **Presentation:** Presentation is a protocol between the reader device and the user.

- $ri_U \leftarrow\$ \mathsf{U.GenEph}(pp)$: The user generates ephemeral input to the reader.

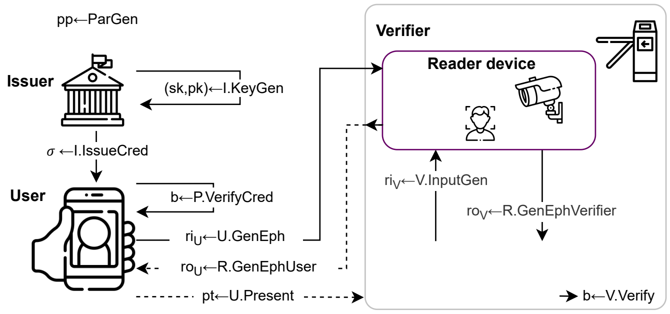

Figure 1. Components and communication flow

- $ro_U \leftarrow\$ \mathsf{R.GenEphUser}(Bio_f, ri_U)$: The reader generates ephemeral output intended for the user from the fresh biometric template.
- $pt \leftarrow\$ \mathsf{U.Present}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a}, \phi, ro_U, ri_U, ctx)$: Prove possession of a credential where the attributes fulfil the predicates, i.e., $\phi(\boldsymbol{a}) = 1$, and the biometric template matches the fresh reading. For the latter, ro_U received from the device will be used. Present returns $\perp$ if the statements are not fulfilled.

d) **Verification:** Verification is an interactive process between the verifier and the reader device.

- $ri_V \leftarrow\$ \mathsf{V.InputGen}(pt)$: Derive from the presentation token the verifier's input to the reader.
- $ro_V \leftarrow\$ \mathsf{R.GenEphVerifier}(Bio_f, ri_V)$: Generate ephemeral public (i.e., intended for verifier) information from the fresh biometric template.
- $b \leftarrow \mathsf{V.Verify}(\mathsf{pk}, pt, \phi, ro_V, ctx)$: Verify the validity of the prover's claims.

Figure 1 illustrates the communication flow of our approach. Design choices leading to this system—like avoiding the use of long-term keys on the device—and their impact are discussed in Section VI. Communications from the reader device to the verifier are assumed to be authentic, which in practice will typically be guaranteed by a physical connection (e.g., via Ethernet) between them.

Following our ambition of minimal assumptions on device capabilities, we only request a secure (i.e., authenticated and encrypted) input channel from the user to the reader device *for a single message*. In practice, this could for instance be realized by using the same input mechanism that is used for biometric recognition, e.g., a camera in case of facial recognition, or using physical proximity, e.g., using near field communication (NFC). Note that without long-term key material on the reader device, this requirement is minimal and cannot be dropped: without such a channel, it would be impossible to avoid a person-in-the-middle action by the verifier, as the user would have no means to verify whether it is communicating with the actual device or a malicious verifier.

In the figure, we display this setting by adding an arrowhead to each entity that can see a message, and continuous lines when the original sender is authenticated. As such, the first message from the user to the reader is represented with a continuous line, and the verifier cannot see the contents of the message. The message coming from the device, however, is routed through the verifier, which can, in principle, see and modify it, as the channel is not authenticated by assumption.

B. Security Model

Next, we define the necessary security properties of a Biometric-Bound Attribute-Based Credentials system.

a) **Correctness:** If all parties follow the protocol specifications, any presentation token generated by the user that is having her biometric measured will be accepted with the same behaviour—i.e. false positive/negative rates—inherent to the biometric matching procedure and its implementation. We omit the formal definition as it is the natural formalization of this property.

b) **Unforgeability:** Unforgeability requires that it is infeasible for an adversary to generate a valid presentation token if it has not previously received a credential satisfying the predicates, i.e., $\phi(\boldsymbol{a}) = 1$, and the biometric matching, i.e., $\mathcal{M}(Bio_f, a_{Bio}) = 1$, or has seen the exact same token. Before outputting a forgery, the adversary is allowed to obtain arbitrarily many presentations of credentials of its choice, and also request credentials on attributes of its choice.

Furthermore, the adversary is given full control over all biometric measurements, but will not win the game if it already asked for a credential that fulfils the predicates and includes a template that matches the fresh biometric template used in the forgery.

Note that our unforgeability notion immediately also covers non-transferability, as presentations are bound to a biometric which is specific to the credential owner.

Definition 1. *A Biometric-Bound Attribute-Based Credentials satisfies* unforgeability *if and only if for every* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that:*

$$\Pr\left[\mathbf{Exp}_{\mathcal{A}}^{Unforgeability}(1^\lambda) = 1\right] \leq \mathsf{negl}(\lambda)\,,$$

where $\mathbf{Exp}_{\mathcal{A}}^{Unforgeability}(1^\lambda)$ *is as defined in Experiment 1.*

c) **Unlinkability:** Unlinkability requires that no adversary can link two user actions when the reader device is honestly executing its protocols. Experiment 2 thus gives the adversary control over the issuer, the user's credential, and the verifier, but not of any process within the reader device. The control over issuance and user credential implies that the adversary will choose the attributes and the biometric templates used in the experiment, albeit with the restriction that no trivial distinction may be possible, i.e., the original templates and fresh biometric scans need to match, the policy needs to be satisfied by the attributes contained in both credentials, and the credentials need to be valid. Giving the adversary full control over the biometrics in particular implies that a scheme proven secure in our model is also secure for any real-world distribution of biometrics.

Definition 2. *A Biometric-Bound Attribute-Based Credentials satisfies* unlinkability *if and only if for every* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that:*

$$\Pr\left[\mathbf{Exp}_{\mathcal{A}}^{Unlinkability}(1^\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)\,,$$

where $\mathbf{Exp}_{\mathcal{A}}^{Unlinkability}(1^\lambda)$ *is as defined in Experiment 2.*

```
$\mathbf{Exp}_{\mathcal{A}}^{Unforgeability}(1^\lambda)$
$pp \leftarrow\$ \mathsf{ParGen}(1^\lambda)$
$Q_{issue} \leftarrow \emptyset, Q_{present} \leftarrow \emptyset, Q_{reveal} \leftarrow \emptyset$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{I.KeyGen}(pp)$
$(pt^*, \phi^*, Bio_f^*, ri_V^*, ctx^*) \leftarrow\$ \mathcal{A}^{\mathsf{O}_{issue}, \mathsf{O}_{present}, \mathsf{O}_{reveal}}(pp, \mathsf{pk})$
  where the oracles are defined as follows:
  $\mathsf{O}_{issue}(j, a_{Bio\,j}, \boldsymbol{a}_j)$
    add $(\{j, a_{Bio\,j}, \boldsymbol{a}_j\})$ to $Q_{issue}$
    $\sigma_j \leftarrow\$ \mathsf{I.IssueCred}(\mathsf{sk}, a_{Bio\,j}, \boldsymbol{a}_j)$
  $\mathsf{O}_{present}(j, \phi, ro_U, ri_U, ctx)$
    add $(\{ro_U, a_{Bio\,j}, \boldsymbol{a}_j, \phi\}, ctx)$ to $Q_{present}$
    return $pt \leftarrow\$ \mathsf{U.Present}(\mathsf{pk}, \sigma_j, a_{Bio\,j}, \boldsymbol{a}_j, \phi, ro_U, ri_U, ctx)$
  $\mathsf{O}_{reveal}(j)$
    add $(\{a_{Bio\,j}, \boldsymbol{a}_j\})$ to $Q_{reveal}$
    return $\sigma_j$
$ro_V^* \leftarrow\$ \mathsf{R.GenEphVerifier}(Bio_f^*, ri_V^*)$
return 1 if:
  $\mathsf{V.Verify}(\mathsf{pk}, pt^*, \phi^*, ro_V^*, ctx^*) = 1 \wedge$
  $(\phi^*(\boldsymbol{a}) = 0 \vee \mathcal{M}(Bio_f^*, a_{Bio}) = 0)\ \forall\{a_{Bio}, \boldsymbol{a}\} \in Q_{reveal} \wedge$
  $\{ro_U^*, {a_{Bio}}^*, \boldsymbol{a}^*, \phi^*, ctx^*\} \notin Q_{present}$
else return 0
```

Experiment 1: Unforgeability experiment

```
$\mathbf{Exp}_{\mathcal{A}}^{Unlinkability}(1^\lambda)$
$b \leftarrow\$ \{0, 1\}$
$pp \leftarrow\$ \mathsf{ParGen}(1^\lambda)$
$(\mathsf{pk}, \phi, \left\{\sigma^t, \boldsymbol{a}^t, {a_{Bio}}^t, Bio_f^t\right\}_{t\in\{0,1\}}, ctx, st) \leftarrow\$ \mathcal{A}(pp)$
$ri_U \leftarrow\$ \mathsf{U.GenEph}(pp)$
$ro_U^* \leftarrow\$ \mathsf{R.GenEphUser}(Bio_f^b, ri_U)$
$ro_U' \leftarrow\$ \mathcal{A}(ro_U^*)$
$pt^* \leftarrow\$ \mathsf{U.Present}(\mathsf{pk}, \sigma^b, {a_{Bio}}^b, \boldsymbol{a}^b, \phi, ro_U', ri_U, ctx)$
$ri_V^* \leftarrow\$ \mathcal{A}(pt^*)$
$ro_V^* \leftarrow\$ \mathsf{R.GenEphVerifier}(Bio_f^b, ri_V^*)$
$b^* \leftarrow\$ \mathcal{A}(pt^*, ro_U^*, ro_V^*, st)$
return 1 if:
  $b = b^*$
  $\mathsf{U.VerifyCred}(\mathsf{pk}, \sigma^t, {a_{Bio}}^t, \boldsymbol{a}^t) = 1,\ t \in \{0, 1\}$
  $\phi(\boldsymbol{a}^t) = 1,\ t \in \{0, 1\}$
  $\mathcal{M}(Bio_f^t, {a_{Bio}}^t) = 1,\ t \in \{0, 1\}$
else return $b \leftarrow\$ \{0, 1\}$
```

Experiment 2: Unlinkability experiment

IV. Our Generic Constructions

In this section we give two generic constructions for the bb-ABC framework, and provide a formal security analysis according to the model described in Section III-B. The key difference between the constructions is the approach to match the freshly captured biometric data with the template contained in the credential.

The first construction in Section IV-A (BioABC-R) performs the matching of the freshly captured biometric data and the template on the reader device. Thus the construction is largely agnostic to the specifics of the respective biometric feature used.

In the second construction in Section IV-B (BioABC-ZK) the matching is performed via a zero-knowledge proof. That is, the user generates a NIZK that the template encoded in the credential and the freshly captured biometric data satisfy the matching algorithm. Due to the requirement of encoding the template into the credential and proving the matching algorithm in zero-knowledge, this puts restrictions on the choice of the biometric features when aiming for practical efficiency. For our instantiation in Section V we will show that facial matching, one natural choice, delivers acceptable performance. A discussion on the choice of ABC-friendly biometric features is deferred to Section VI.

A. BioABC-R: Matching on Reader

Our first construction, BioABC-R depicted in Construction 1, can be built from a digital signature scheme Σ, a non-interactive zero-knowledge proof of knowledge protocol Π, and an authenticated encryption scheme $\mathcal{E}$. There, the template matching is performed inside the reader device. The user computes a commitment to her biometric template a_{Bio}, and encrypts it together with the opening of the commitment such that only the device can access the template in plain. As part of the presentation token, she proves in zero-knowledge that the template in the credential corresponds to the committed value. The verifier then checks the proof, and defers to the device for the matching result, including a check that the encrypted value used for the matching algorithm is the opening of the commitment used in the proof.

For the sake of notational simplicity, we directly encode the biometric template into the credential; however, it is straightforward to alternatively encode a hash value in the credential (and disclose it upon presentation), and let the reader device also check the correctness of the hash value. This makes the number of attributes in the credential fully independent of the actual biometrics scheme being used, which may be a benefit, e.g., in case of very large or high-dimensional templates.

Theorem 1. *If Σ is* EUF-CMA*-secure, Π is zero-knowledge and simulation-sound extractable and $\mathcal{C}$ is computationally binding, then Construction 1 is unforgeable.*

Proof Sketch. We only include an informal description of the ideas underlying the proof here, and refer to Appendix A.A for the full proof.
The proof relies on the zero-knowledge and simulation-sound extractability properties of Π to obtain a witness from the forgery of an adversary that breaks the unforgeability game, which is then argued to lead to a forgery for Σ in the EUF-CMA experiment, which has negligible probability. A subtlety arises on this argument, due to the introduction of biometric templates. It is necessary to remove the possibility of the adversary trying to use a different biometric template for the forgery to get an advantage. The binding property of the commitment scheme avoids possible modifications of the value committed for the zero-knowledge proof. Lastly, the trusted reader device is in charge of matching the templates, and checking whether the matched template was actually the one committed to by the user through the R.GenEphVerifier method. □

$\underline{\mathsf{ParGen}(1^\lambda)}$. Get $pp' \leftarrow\$ \ \Sigma.\mathsf{ParGen}(1^\lambda)$. Get $pp'' \leftarrow\$$ $\mathcal{E}.ParGen(1^\lambda)$. Return $pp = \{pp', pp''\}$.

Key Generation and Issuance.

$\underline{\mathsf{I.KeyGen}(pp)}$. Return $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \ \Sigma.\mathsf{KeyGen}(pp')$

$\underline{\mathsf{I.IssueCred}(\mathsf{sk}, a_{Bio}, \boldsymbol{a})}$. Set $\boldsymbol{b} = (a_{Bio}, \boldsymbol{a})$. Return $\sigma \leftarrow\$ \ \Sigma.\mathsf{Sign}(\mathsf{sk}, \boldsymbol{b})$

$\underline{\mathsf{U.VerifyCred}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a})}$. Set $\boldsymbol{b} = (a_{Bio}, \boldsymbol{a})$. Return 1 if $\Sigma.\mathsf{Verify}(\mathsf{pk}, \boldsymbol{b}, \sigma) = 1$, else return 0

Presentation.

$\underline{\mathsf{U.GenEph}(pp)}$. $(\mathsf{sk}_{ae}) \leftarrow\$ \ \mathcal{E}.\mathsf{KeyGen}(pp')$. Return $ri_U = \mathsf{sk}_{ae}$.

$\underline{\mathsf{R.GenEphUser}(Bio_f, ri_U)}$. Parse and store $ri_U = \mathsf{sk}_{ae}$.

$\underline{\mathsf{U.Present}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a}, \phi, ro_U, ri_U, ctx)}$. Compute $(C_{a_{Bio}}, V_{a_{Bio}}) \leftarrow\$ \ \mathcal{C}.\mathsf{Commit}(a_{Bio})$, and $ae_{a_{Bio}} \leftarrow\$ \ \mathcal{E}.\mathsf{Encrypt}(\mathsf{sk}_{ae}, \{V_{a_{Bio}}, a_{Bio}\})$. Run:

$$\begin{aligned} pt' \leftarrow\$ \ \mathsf{NIZK}[(\sigma, a_{Bio}, V_{a_{Bio}}, \boldsymbol{a}) :& \\ &\mathsf{U.VerifyCred}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a}) = 1 \wedge \\ &\mathcal{C}.\mathsf{Open}(C_{a_{Bio}}, V_{a_{Bio}}, a_{Bio}) = 1 \wedge \\ &\phi(\boldsymbol{a}) = 1](\phi, ctx) \end{aligned}$$

Return $pt = \{ae_{a_{Bio}}, C_{a_{Bio}}, pt'\}$

Verification.

$\underline{\mathsf{V.InputGen}(pt)}$. Parse pt as $\{ae_{a_{Bio}}, C_{a_{Bio}}, pt'\}$ and return $ri_V = \{ae_{a_{Bio}}, C_{a_{Bio}}\}$.

$\underline{\mathsf{R.GenEphVerifier}(Bio_f, ri_V)}$. Parse $ri_V = \{ae_{a_{Bio}}, C_{a_{Bio}}\}$. Compute $\{a_{Bio}, V_{a_{Bio}}\} = \mathcal{E}.\mathsf{Decrypt}(\mathsf{sk}_{ae}, ae_{a_{Bio}})$. If decryption fails, output $\perp$. If $\mathcal{C}.\mathsf{Open}(C_{a_{Bio}}, V_{a_{Bio}}, a_{Bio}) = 0$, return 0. Return the result of $\mathcal{M}(a_{Bio}, Bio_f)$.

$\underline{\mathsf{V.Verify}(\mathsf{pk}, pt, \phi, ro_V, ctx)}$. Parse $b \leftarrow ro_V$. If 0 return 0. Otherwise, parse $pt = \{ae_{a_{Bio}}, pt'\}$. Return the verification result of pt'

Construction 1: BioABC-R

Theorem 2. *If Π is zero-knowledge, $\mathcal{E}$ is an authenticated encryption scheme and $\mathcal{C}$ is a computationally hiding commitment scheme, then Construction 1 is unlinkable.*

Proof Sketch. In the proof, we perform a series of modifications of the unlinkability experiment supported by the building block's security properties, as well as the construction's procedures. We only include an informal description of the ideas behind the proof here, and refer to Appendix A.B for the full proof.

First, we return a simulated proof instead of running U.Present, and the result will be computationally indistinguishable for the adversary due to Π being zero-knowledge.
Then we can modify the game (due to the authenticated encryption) to remove the adversary's control over ro_U^*, that is, to let the challenger receive directly the result coming from R.GenEphUser for its computations. In this construction the value of ri_V^* is an encrypted value and a commitment. As the encryption is authenticated, and the commitment scheme hiding, the control of the adversary over that value can be

removed (its changes would be detected by $\mathsf{R.GenEphVerifier}$ except for a negligible probability). What is more, we can further modify the experiment so the adversary does not even receive the honest ri_V^* value, because of indistinguishability of the encryption scheme $\mathcal{E}$ and the hiding property of the commitment scheme $\mathcal{C}$. After these modifications, the resulting experiment does not give the adversary any input related to the chosen bit b, which concludes the proof. □

B. BioABC-ZK: Matching via ZK Proof

The previous construction fulfils the security model as defined in Section III-B. However, it puts high trust assumptions on the reader device, as it is in charge of doing the actual biometric matching. One of the consequences is that users cannot know whether they were evaluated fairly or not. If they are rejected access, they cannot be sure whether the reader device responded with an honest matching result that lead to a false negative (which may happen in any biometric authentication scheme) or not. What is more, they cannot know whether the verifier decided to ignore the reader's result and simply denied them access.

In this section, we introduce a new construction that adds auditing capabilities to the user and the overall system, adding an extra layer of confidence and a reduction of necessary trust in the overall setup. Construction 2 introduces an actual proof in zero-knowledge by the user of the matching between fresh and credential biometric templates. Every user device can monitor whether the behaviour of the reader matches the expected, thus detecting whether the reader behaves inconsistently and/or the verifier decided incorrectly without losing privacy guarantees. Note that in this construction it is not only possible to check expected false positive and negative rates, but the template the user receives gives extra information to detect a forged template (e.g., noting a statistically unlikely similarity value). Malicious behaviour can also be provably demonstrated to other parties. As will be further exemplified in the next section, the trade-off is a loss of efficiency, which now depends on the specific biometric matching method used.

Given a digital signature scheme Σ, a non-interactive zero-knowledge proof of knowledge protocol Π, a commitment scheme $\mathcal{C}$ and an authenticated encryption scheme $\mathcal{E}$ as defined in Section II-C, the BioABC-ZK construction is depicted in Construction 2. The user receives from the reader device the fresh biometric data, along with a commitment to it and the opening value, while the verifier only gets the commitment. Note that the sensitive information will be protected through the encryption with a fresh key generated by the user, so it will not be accessible to any other party. The user can then prove in zero-knowledge to the verifier that the biometric data within her credential matches the fresh template, along with the rest of the statements defined by ϕ.

Theorem 3. *If Σ is* EUF-CMA*-secure, Π is zero-knowledge and simulation-sound extractable, and $\mathcal{C}$ is computationally binding, then Construction 2 is unforgeable.*

The full proof of this theorem can be found in Appendix A.C. The ideas are very similar to the unforgeability proof of BioABC-R, with the key difference being the way the use of different templates for a forgery is ruled out.

$\underline{\mathsf{ParGen}(1^\lambda)}$. Get $pp' \leftarrow\$ \Sigma.\mathsf{ParGen}(1^\lambda)$. Get $pp'' \leftarrow\$ \mathcal{E}.ParGen(1^\lambda)$. Return $pp = \{pp', pp''\}$.

Key Generation and Issuance.

$\underline{\mathsf{I.KeyGen}(pp)}$. Return $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \Sigma.\mathsf{KeyGen}(pp')$

$\underline{\mathsf{I.IssueCred}(\mathsf{sk}, a_{Bio}, \boldsymbol{a})}$. Set $\boldsymbol{b} = (a_{Bio}, \boldsymbol{a})$. Return $\sigma \leftarrow\$ \Sigma.\mathsf{Sign}(\mathsf{sk}, \boldsymbol{b})$

$\underline{\mathsf{U.VerifyCred}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a})}$. Set $\boldsymbol{b} = (a_{Bio}, \boldsymbol{a})$. Return 1 if $\Sigma.\mathsf{Verify}(\mathsf{pk}, \boldsymbol{b}, \sigma) = 1$, else return 0

Presentation.

$\underline{\mathsf{U.GenEph}(pp)}$. $(\mathsf{sk}_{ae}) \leftarrow\$ \mathcal{E}.\mathsf{KeyGen}(pp')$. Return $ri_U = \mathsf{sk}_{ae}$.

$\underline{\mathsf{R.GenEphUser}(Bio_f, ri_U)}$. Parse $ri_U = \{\mathsf{sk}_{ae}\}$. Compute $(C_{Bio_f}, V_{Bio_f}) \leftarrow\$ \mathcal{C}.\mathsf{Commit}(Bio_f)$. Return $ro_U = \mathcal{E}.\mathsf{Encrypt}(\mathsf{sk}_{ae}, \{C_{Bio_f}, V_{Bio_f}, Bio_f\})$.

$\underline{\mathsf{U.Present}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a}, \phi, ro_U, ri_U, ctx)}$. Parse $\{C_{Bio_f}, V_{Bio_f}, Bio_f\} = \mathcal{E}.\mathsf{Decrypt}(\mathsf{sk}_{ae}, ro_U)$. If decryption fails, return $\perp$. Return

$$
\begin{aligned}
pt \leftarrow\$ \mathsf{NIZK}[(\sigma, a_{Bio}, \boldsymbol{a}, Bio_f, V_{Bio_f}) : \\
&\mathsf{U.VerifyCred}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a}) = 1 \wedge \\
&\mathcal{M}(a_{Bio}, Bio_f) = 1 \wedge \\
&\mathcal{C}.\mathsf{Open}(C_{Bio_f}, V_{Bio_f}, Bio_f) = 1 \wedge \\
&\phi(\boldsymbol{a}) = 1](\phi, ctx)
\end{aligned}
$$

Verification.

$\underline{\mathsf{V.InputGen}(pt)}$. Return $ri_V = \varepsilon$.

$\underline{\mathsf{R.GenEphVerifier}(Bio_f, ri_V)}$. Return C_{Bio_f} as computed in $\mathsf{R.GenEphUser}$

$\underline{\mathsf{V.Verify}(\mathsf{pk}, pt, \phi, ro_V, ctx)}$. Parse $C_{Bio_f} \leftarrow ro_V$ and use it for verification of proof. If pt verifies correctly return 1. Else, return 0

Construction 2: BioABC-ZK

Theorem 4. *If Π is zero-knowledge, $\mathcal{C}$ is computationally hiding, and $\mathcal{E}$ is an authenticated encryption scheme, then Construction 2 is unlinkable.*

We refer to Appendix A.D for the full proof of this theorem, which uses ideas similar to the unlinkability proof of BioABC-R.

V. Instantiation

In this section, we delve into practical instantiations of the generic constructions, and evaluate their security and efficiency.

A. BioABC-R Instantiation

The BioABC-R construction leaves the burden of matching the biometric template completely to the reader device. This simplifies the requirements on the building blocks for the construction, as the complexity of the biometric matching does not affect the cryptographic primitives. In fact, the only requirement for the proof is that we can link the template to the value in the credential, which can actually be done via hashing the biometric template.

More specifically, in this construction we can sign the hashed biometric template, create a commitment that works in a hash-and-commit way (for instance a Pedersen commitment to the hashed template), do the proof on the committed value, and give the reader device the actual biometric template so it can check that its hash corresponds to the committed value.

For the credential generation and proof, any current ABC scheme, like Pointcheval-Sanders signatures (PS) [49]. This will cover both the signature and zero-knowledge proof building blocks, as the proof only needs being able to hide the biometric attribute and prove a commitment to it (akin to inspection), which is supported by many credential schemes. For authenticated encryption, we may choose authenticated AES-GCM-256 [50] as it fulfills the needed security properties.

The impact on efficiency of this biometric-bound version over simple ABCs is almost zero. Indeed, the overhead over a presentation where the user proves the fulfillment of the predicates (e.g., being vaccinated or recovered) is just adding an extra attribute during the presentation (the hash of the biometric template), a single authenticated encryption/decryption of the biometric template, and a commitment opening check.

B. BioABC-ZK Instantiation

In the BioABC-ZK construction, the matching of the biometric templates has to be proved in zero-knowledge. For practical applications, this puts a limitation on the biometrics that may be used. This topic will be further discussed in Section VI. In our instantiation, we will focus on facial matching, which is the decision problem of whether two face pictures belong to the same person or not.

In this field, there are multiple works based on extracting a vector template from each picture, and comparing them with a similarity measurement, e.g., [41], [51]. In particular, our solution is based on the matching system presented by Ouamane et al. [42]. The method consists on feature extraction followed by dimensionality reduction techniques that lead to a facial biometric template. The templates of two pictures are compared through the cosine similarity metric: $\frac{\langle \boldsymbol{x},\boldsymbol{y}\rangle}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$. If the value is over a threshold τ, the result is a positive match. With this approach, the system gets up to 95% accuracy using a template with $N = 600$ components. These results were obtained using the Faces in the Wild[4] database, which is a realistic setting for our scenarios, where there would be no hard restrictions on users when sampling their biometrics.

To overcome the issue of floating point arithmetic, we rewrite the condition $\frac{\langle \boldsymbol{x},\boldsymbol{y}\rangle}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} \geq \tau$ to the equivalent form $\left\langle 2^l \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}, 2^l \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|} \right\rangle \geq 2^{2l}\tau$, thereby turning cosine similarity into an inner product statement, which can efficiently be proven in zero-knowledge. As such, we represent the decimal values of the templates as $\mathbb{Z}_p$ elements for the computations. We use encodings of $l = 100$ bits, which offers high precision, yet avoids potential overflows in the computation if the values of templates' components are trusted. Note that this does not actually add any new trust assumptions in the system: issuers were already trusted to only sign correct biometric templates, and also reader devices have to be trusted to generate correct templates.

$\underline{\mathsf{ParGen}(1^\lambda)}$. Return $pp \leftarrow\$ \, PS.\mathsf{ParGen}(1^\lambda)$

Key Generation and Issuance.

$\underline{\mathsf{I.KeyGen}(pp)}$. Return $(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \, PS.\mathsf{KeyGen}(pp)$

$\underline{\mathsf{I.IssueCred}(\mathsf{sk}, a_{Bio}, \boldsymbol{a})}$. $\boldsymbol{b} = (a_{Bio}, \boldsymbol{a})$. Return $\sigma \leftarrow\$ \, PS.\mathsf{Sign}(\mathsf{sk}, \boldsymbol{b})$

$\underline{\mathsf{U.VerifyCred}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a})}$. $\boldsymbol{b} = (a_{Bio}, \boldsymbol{a})$. Return 1 if $PS.\mathsf{Verify}(\mathsf{pk}, \boldsymbol{b}, \sigma) = 1$, else return 0

Presentation.

$\underline{\mathsf{U.GenEph}(pp)}$. Return $\mathsf{sk}_{ae} \leftarrow\$ \, AES.\mathsf{KeyGen}()$.

$\underline{\mathsf{R.GenEphUser}(Bio_f, ri_U)}$. Parse $ri_U = \{\mathsf{sk}_{ae}\}$. Compute $(C_{Bio_f}, V_{Bio_f}) \leftarrow\$ \, BitPC.\mathsf{Commit}(Bio_f)$, i.e., $C_{Bio_f} = (C_1, \ldots, C_N)$ to the individual bits f_i of Bio_f, and $V_{Bio_f} = (r_1, \ldots, r_N)$ contains the individual openings. Return $ro_U = AES.\mathsf{Encrypt}(\mathsf{sk}_{ae}, \{C_{Bio_f}, V_{Bio_f}, Bio_f\})$.

$\underline{\mathsf{U.Present}(\mathsf{pk}, \sigma, a_{Bio}, \boldsymbol{a}, \phi, ro_U, ri_U, ctx)}$. Parse $\{C_{Bio_f}, V_{Bio_f}, Bio_f\} = AES.\mathsf{Decrypt}(\mathsf{sk}_{ae}, ro_U)$. If decryption fails, return $\perp$. $\boldsymbol{b} = (a_{Bio}, \boldsymbol{a})$.

Given $a_{Bio} = (e_i)_{i\in[N]}$, $Bio_f = (f_i)_{i\in[N]}$. Choose random blinding values $w, z \leftarrow\$ \, \mathbb{Z}_r$. Take σ as (a', σ_1, σ_2) and compute $(\sigma_1', \sigma_2') = (\sigma_1^w, (\sigma_2\sigma_1^z)^w)$. Then, compute an Schnorr-style proof:

$pt \leftarrow\$ \, \mathsf{NIZK}[(\sigma, (e_i), \boldsymbol{a}, (f_i), (r_i), s, r) :$

PS credential check $$e(g_1^t X \prod_{i=1}^{N} Y_{e_i}^{e_i} \prod_{j=1}^{n} Y_{a_j}^{a_j} Y_{k+1}^{a'}, \sigma_1') = $$
$$= e(g_1, \sigma_2') e(X, \sigma_1')^{-1} \wedge$$

Valid commitment $$\bigwedge_{i=1}^{N} C_i = g^{f_i} h^{r_i} \wedge$$

Valid inner product $s = \langle e, f \rangle$ $$1 = \prod_{i=1}^{N} C_i^{e_i} g^{-s} h^{-r} \wedge$$

Biometric match $s \in [2^{2l}\tau, 2^{2l}] \wedge$

Predicate check $\phi(\boldsymbol{a}) = 1](\phi, ctx)$

Verification.

$\underline{\mathsf{V.InputGen}(pt)}$. $ri_V = \varepsilon$.

$\underline{\mathsf{R.GenEphVerifier}(Bio_f, ri_V)}$. Return C_{Bio_f} as computed in $\mathsf{R.GenEphUser}$

$\underline{\mathsf{V.Verify}(\mathsf{pk}, pt, \phi, ro_V, ctx)}$. Parse $C_{Bio_f} \leftarrow ro_V$ and use it for verification of proof. If pt verifies correctly return 1. Else, return 0

Construction 3: Concrete instantiation of BioABC-ZKs

Construction 3 shows the instantiation of the BioABC-ZK construction with the facial biometric method [42], using a template length of $N = 600$. For instantiating ABCs, we use Pointcheval-Sanders signatures (PS) [49] in a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, g_1, g_2)$, which allow zero-knowledge showings. Additionally, we rely on Pedersen commitments (PC) [44] and authenticated AES-GCM-256 (AES) [50]. The presentation token is a Schnorr-style proof of knowledge (Σ-protocol) turned non-interactive using the Fiat-Shamir heuristic [52] which gives us a simulation-sound extractable NIZK proof [53]. The statement and public values are included in the computation of the challenge in order to avoid malleability issues [54], and the context includes information that avoids replay attacks (e.g., current time).

Corollary 1. *The instantiation presented in Construction 3 is unforgeable and unlinkable.*

[4] http://vis-www.cs.umass.edu/lfw/results.html

Proof. The security properties follow from the unforgeability and unlinkability of the generic construction. PS credentials as presented in [49] are EUF-CMA-secure in type-3 bilinear groups in the random oracle model and under a variant of the q-SDH assumption, which was shown to hold on the generic bilinear group model. Additionally, Pedersen commitments are computationally binding under the discrete logarithm assumption. Lastly, the required Σ-protocols when used with Fiat-Shamir yield simulation-sound extractable NIZK proofs [53]. AES-256 is an authenticated encryption scheme, Pedersen commitments are perfectly hiding, and, as shown before, the instantiation of the presentation protocol is zero-knowledge. Therefore, the BioABC-ZK instantiation is also unlinkable. □

1) Micro-Benchmark: We next give feasibility micro-benchmarks for the BioABC-ZK instantiation. Our focus is on the overhead such a scheme would have over a simple credential showing. Thus, we measured values for the expensive tasks executed by each of the actors during a presentation phase. Namely, this entails the computation of Pedersen commitments in the reader device, and the tasks related to the zero-knowledge proof for user and verifier.

As shown in Construction 3, the NIZK involves five main statements: checking the validity of the credential, checking the validity of the commitments, proving the inner product computation, proving the matching condition (a range proof), and checking the predicates over the attributes. The latter would be dependent on the access policy, and corresponds to the computations in a traditional showing. Further, for predicate proving the attributes are linked to the credential through commitments and predicate proofs are done over those commitments. Therefore, that part of the proof is independent from the rest, and the overhead of our instantiation is independent of the complexity of the predicates. Because of this, our analysis will be centred on the rest of the computations, presenting fine-granular timings to get a better picture on the complexity and possible optimization points. Note that, while credential validation would also need to be done in a traditional showing, including the costly pairing computations, we consider the whole computation as overhead because of the significant increase in the number of attributes due to the biometric template ($N = 600$).

Specifically, we focus on the following operations:

- PS credential validation with 600 attributes requires about 600 multi-exponentiation plus 1 pairing operation.
- Proving that the 600 Pedersen commitments are valid.
- Proving the value of the inner product in zero-knowledge. This implies a 602 multi-exponentiation, cf. Construction 3.
- Proving that the biometric templates match. This is, in fact, a range proof. As the user is more constrained, we consider a simple bit-by-bit decomposition proof for this, resulting in larger proof sizes than advanced techniques, but minimizing the user costs for the given parameters. This proof requires 200 Pedersen commitments of bits (that is, they can actually be computed with 200 exponentiations), 200 exponentiations for the “real” OR branches, and 200 Pedersen commitments for the simulated OR branches.
- The reader device has to compute 600 Pedersen commitments for the fresh template.

For the benchmarks, we use the parameters (i) $N = 600$ components for the biometric template, as suggested in [42], (ii) l=100 bits for template representation, (iii) 48 byte representation of $\mathbb{Z}_p$ elements, as used in our implementation, (iv) 97 byte representation of elements from the source group $\mathbb{G}_1$ of the bilinear group, as used in our implementation. For time measurements, the mean over 20 repetitions of the computations was taken, with 30 warm-up iterations.

a) **Demonstration Setup:** In our setting, the user and reader device are mobile and embedded, respectively, while the verifier can be assumed to be more powerful (e.g., a normal computer). To reflect this, we took timing values in different devices. As *user*, we used a Poco X3 NFC with a Qualcomm Snapdragon 732G octa-core 2.3GHz. The timings for the *reader* were taken on a Raspberry Pi 3 Model B, with an ARM-Cortex-A53m, 1.2 GHz. Lastly, the *verifier*’s results come from executing benchmarks in a GF63 Thin 95C laptop with Intel Core i7-9750H CPU, 2.60GHz. In all cases, the implementation was based on a C project using the Miracl Core[5] library for elliptic-curve operations, concretely on the pairing-friendly BLS12-381 curve.

Table I shows the results of the experiments. The total overhead is just over 3 seconds. However, as this is a feasibility result, we only performed one key optimization—establishing lookup tables for the 2^j powers of the commitment bases g, h, reducing execution time of commitment computations to around half the time at the cost of $\approx 35KB$ of memory—and there is still room for other optimizations, like using algorithms that take advantage of the 2-exponentiation structure of Pedersen commitments, implementations tailor-made for the constrained devices, etc. What is more, we remark the possibility of the following precomputations being carried out, as marked in the corresponding column in Table I:

- As the credential validity proof operations (multi-exponentiation and pairing) do not depend on the fresh values, they can be fully precomputed. For instance, the process can be started the moment the user application is opened in the mobile phone in a practical scenario.
- For the reader device, the randomness in the Pedersen commitments (h^{r_i}) can be precomputed during idle time (between readings), halving the *online* execution time.
- The operations for the matching biometric proof could be precomputed, at the cost of doubling the actual computation time. Indeed, the bit-by-bit range proof involves an OR proof on the bits. For each bit, the user could compute the proofs for both possible cases, and only send the correct one once the inner product value is known. The usefulness of this precomputation would depend heavily on the specific implementation and use case characteristics.

Even assuming that the last optimization is not available in a specific scenario, applying the other two would lead to an exe-

[5] https://github.com/miracl/core

Entity	Process	Time (s)	Precomputable	Total (s)
User	PS cred	0.149	Yes	1.103
	Pedersen	0.463	No	
	Inner product	0.119	No	
	Bio match	0.372	Yes*	
Verifier	PS cred	0.060	No	0.415
	Pedersen	0.176	No	
	Inner product	0.044	No	
	Bio match	0.135	No	
Reader	Pedersen	1.677	Partially	1.677
Total	All processes without precomputation			3.195

Table I
TIMING RESULTS FOR THE OVERHEAD COMPUTATIONS.

cution time of ≈ 2.1 seconds, which could be further reduced by applying more complex optimizations. This demonstrates the feasibility of the solution in practical applications. Note that in scenarios where the waiting time is even more critical, it would be possible to apply the more efficient solution based on the BioABC-R construction. An additional overhead is the AES encryption/decryption of $1'200$ $\mathbb{Z}_p$ elements–that is, the fresh template vector plus the randomness–and 600 $\mathbb{G}_1$ elements–the commitments. This results in a total of $115'800$ bytes, which would lead to a few more milliseconds at most, as current implementations usually achieve a throughput of more than hundreds of Mb per second.

We remark that the communication complexity is not prohibitive. Indeed, as mentioned in the previous paragraph, the reader would need to send around 115KB of data (then encrypted with AES) to the user, and 58KB–the template commitments–to the verifier. Additionally, the size of the proof sent by the user will be 2 $\mathbb{G}_1$ elements, plus $1'800$ $\mathbb{Z}_p$ elements for the witnesses and an extra 200 $\mathbb{G}_1$ and 600 $\mathbb{Z}_p$ elements for the bit-by-bit range proof. Thus, the total proof size would be below 135KB.

VI. DISCUSSION

In the following, we briefly discuss some design choices, limitations, natural extensions, and possible future directions for bb-ABC.

A. System Aspects

a) **Reader device assumptions:** Some kind of trust assumption is inherent to biometrics. In our framework and constructions, we aimed for minimizing it by only requiring a small component with reduced functionality to be audited and certified. This is significantly cheaper and easier than doing it for the whole verifier and also allows for re-using device certifications for different verifier entities. For its integration in the system, besides the natural assumption of an authentic (e.g., physical) connection to the verifier, we only assume a single secure input from the user to the verifier, which, as discussed in Section III, cannot be avoided. We designed our constructions with characteristics that help justify the necessary trust on the reader device from users and verifiers. First, we do not require readers to store long term keys, thereby minimizing the attack surface. Additionally, the reduced functionality of the device simplifies audit and certification processes. Also, a compromised verifier–which is more probable because of their complexity and heterogeneity–does not easily lead to compromising the device because of its reduced communication needs.

b) **Deployment aspects:** Another goal considered for our constructions was reducing deployment complexity. Verifying signatures within the trusted device would require deploying certificates (which might differ per e.g., country or application) to this device, and rotating and updating them via firmware updates. Our constructions rely only on commitments and fresh ephemeral keys. The required public commitment parameters can be set during production and do not need to be changed. Additionally, the minimal hardware requirements for devices facilitate easy adoption.

c) **Long-term security:** All instantiations proposed in this paper fulfill the security properties established for bb-ABC. However, long-term security, specifically in regards to a post-quantum scenario, is desirable, especially in terms of privacy: in particular when processing sensitive data like biometric templates, transcripts need to be protected from deanonymization also in the mid to far future.

While our constructions do not offer post-quantum unforgeability (e.g., through breaking of the signature scheme), we still want to stress that both proposed instantiations achieve long-term unlinkability: Indeed, Pedersen commitments are perfectly hiding, and we use perfect honest-verifier zero-knowledge proofs with the Fiat-Shamir transform for constructing the NIZK proof. Additionally, the best known attack on AES-GCM-256 in realistic quantum models uses Grover's algorithm and leads to a square root improvement, preserving 128 bit of security.

d) **User trust:** We are aware that users' perceived trust on the solution will be important for real-world adoption. This common issue for privacy-enhancing technologies requires solutions typically outside the scope of the technology itself. It must rely on educational campaigns or other societal approaches for trust building.

B. Additional ABC Features

a) **Inspection and non-frameability:** One common feature of ABC systems to offer a trade-off between anonymity for honest users and accountability for misbehaving users is inspection [5], [48]. Here one introduces a dedicated entity (the *inspector*) who can recover certain information about the holder from a transcript of a credential showing, e.g., some specific attributes of the user or her precise identity. Typically this is done by encrypting some attributes under the public key of the inspector and proving consistency of those attributes with the credential in zero-knowledge as part of the presentation. We note that both our constructions support inspection out of the box if one instantiates the commitment scheme with a public-key encryption scheme in a way that only the inspector holds the corresponding secret key. In case that additionally non-frameability [55] is required, issuance could be turned into an interactive protocol where the user embeds a secret attribute, for which every presentation will additionally prove knowledge.

b) **Revocation:** Revocation allows to invalidate issued credentials and can generically be added [11] by including an additional attribute acting as an identifier. Consequently, this can be easily added to our framework in Section III based on the most suitable revocation approach for a given application setting.

c) **User secret and pseudonyms:** As mentioned above, in certain scenarios it might be required to include an additional user secret as an attribute into the credential that is not known to the issuer. This is of particular interest when this additional secret is used to derive scope exclusive pseudonyms [11], so that users per scope (i.e,. application context) deterministically derive a pseudonym from their user secret to realize the feature of controlled linkability, i.e., all showings of a user within a scope are linkable but the user still remains anonymous.

C. Biometric Features for BioABC-ZK

Biometric extraction has inherent noise and fuzziness such as rotations, translations or non-linear deformations (e.g., plasticity of the skin, or changes in luminosity). Consequently, even for the same person, multiple readings will lead to slightly different results.

As discussed in Section I, mitigating this challenge by generating deterministic outputs from a person's biometric readings, e.g., using fuzzy extractors, is not yet practical [31]. Thus, we need to be able to connect biometric matching algorithms, i.e., deciding whether two readings are close enough to belong to the same person, with zero-knowledge proofs. However, this puts a constraint on which biometric systems can be used.

For instance, *fingerprints* are among the most widespread biometric identification systems. State of art fingerprint matching solutions (and most throughout its history) are based on minutiae (local ridge features) extraction. The extracted minutiae of two fingerprints are then matched and compared, typically following a procedural approach: first, the matching probabilities between all minutiae in both fingerprints are computed, and a final matching is established from those probabilities. Then, the similarity score is computed from the aligned templates [56]. This kind of procedure is not translatable into efficient ZK knowledge proofs with existing techniques, in particular in terms of prover complexity.

Other biometrics are compatible with ZK proofs but would not be practical in current scenarios because of complexity parameters, namely those that affect proof size and execution times. This is the case of *iris recognition*, a biometric surging in popularity[6]. Iris recognition is performed through the matching of iris binary codes using the Hamming distance, which fits with ZK proofs as shown in Adam's proposal [19]. However, the codes in practical matching systems are comprised of more than $10'000$ bits. Even when slightly weakening our privacy definition (to account for leaking the rotation of the code due to slight pose differences in the scanning phase), this means that for the ZK showings more than $20'000$ commitments are needed for current approaches, leading to (non-optimized) execution times over 20 seconds and proof sizes of more than 1MBs (estimated using the same values for base operations and sizes as in Section V). While nearing practical values, these constraints would still be too high for real world scenarios, though in the future it could be an interesting system because of its high accuracy levels. In a nutshell, Construction 2 can be instantiated practically with schemes that use simple metrics for comparison (e.g., cosine similarity) and small enough templates (which depends on general computing power of devices, currently fewer than 1000 elements lead to reasonable times as shown in our benchmark).

Lastly, while out of scope of this paper, liveness detection is an important topic in biometrics. We note that this step could be performed by the biometric reader before returning any output about the biometric reading. In fact, both active (e.g., requesting the user to blink) and passive methods would be possible. The integration of liveness detection is thus seamless, but, as in any other system, it would affect the complexity of the procedure.

D. Reducing Privacy Risks

In traditional biometric authentication systems there are some issues due to the nature of biometrics. Firstly, if a biometric feature is used across several different systems, it can be used for linking individuals, and, secondly, if a biometric feature is compromised, it cannot be revoked. One approach to counter such problems is the use of so called cancelable biometrics [57], where in contrast to the direct use of a biometric feature one applies an intentional, repeatable, and non-invertible distortion based on a chosen transform to the biometric signal (on template generation as well as measurement). We note that the approach of cancelable biometrics can equivalently be applied in the setting of bb-ABC.

VII. Conclusion

Biometric-Bound Attribute-Based Credentials are an interesting tool for privacy-preserving physical access control, but are a largely unexplored field when it comes to their practical use. While there has been conceptual work in this direction, in this paper we are the first to rigorously formalize this concept and present performance figures for a practical instantiation based on a concrete biometric feature. Although we consider this an important step towards their real-world use, there remain numerous aspects that deserve further study. First, it would be interesting to investigate the practicality of such a system based on a full implementation and deployment of the system with all the different actors. Secondly, the study of ABC—and zero-knowledge (ZK)—friendliness of other biometric features and matching algorithms is an interesting avenue. For instance, other ZK proof systems such as zk-SNARKs allow to handle the respective matching algorithms in a more natural way. However, they are usually not very "prover-friendly" and the costly computations required by the user might be prohibitive. Nevertheless, there might be interesting trade-offs that can be explored. Thirdly, as discussed in Section VI-A our concrete instantiation provides long-term

[6] E.g., iris scans are used as part of India's identification programme, https://uidai.gov.in/

privacy which even holds when the adversary has access to a powerful quantum computer. However, unforgeability in such a setting is clearly lost. Consequently, an interesting avenue is to investigate the possibility of (practical) fully post-quantum secure schemes.

Acknowledgments

The authors would like to thank Antonio Skarmeta and Sebastian Ramacher for insightful discussions and feedback on earlier versions of this work. This work was in part funded by the European Union under the H2020 Programme Grant Agreement No. 830929 (CyberSec4Europe), the H2020 Programme Grant Agreement No. 883335 (ERATOSTHENES), the Horizon Europe Programme Grant Agreement No. 210808104 (SUNRISE), and Grant PID2020-112675RB-C44 funded by MCIN/AEI/10.13039/5011000011033. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[2] ——, "Security without identification: Transaction systems to make big brother obsolete," *Commun. ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[3] S. Brands, "Rethinking public key infrastructure and digital certificates – building in privacy," Ph.D. dissertation, 1999, phD thesis.

[4] C. Paquin and G. Zaverucha, "U-prove cryptographic specification v1.1 (revision2)," April 2013, technical report, Microsoft Corporation.

[5] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *EUROCRYPT 2001*. Springer, 2001.

[6] ——, "A signature scheme with efficient protocols," in *SCN 2002*. Springer, 2002.

[7] ——, "Signature schemes and anonymous credentials from bilinear maps," in *CRYPTO 2004*. Springer, 2004.

[8] J. Camenisch and E. V. Herreweghen, "Design and implementation of the *idemix* anonymous credential system," in *ACM CCS 2002*. ACM, 2002, pp. 21–30.

[9] U. Haböck and S. Krenn, "Breaking and fixing anonymous credentials for the cloud," in *CANS 2019*. Springer, 2019.

[10] J. Bobolz, F. Eidens, S. Krenn, S. Ramacher, and K. Samelin, "Issuer-hiding attribute-based credentials," in *CANS 2021*. Springer, 2021.

[11] J. Camenisch, S. Krenn, A. Lehmann, G. L. Mikkelsen, G. Neven, and M. Ø. Pedersen, "Formal treatment of privacy-enhancing credential systems," in *SAC 2015*. Springer, 2015.

[12] L. Hanzlik and D. Slamanig, "With a little help from my friends: Constructing practical anonymous credentials," in *ACM CCS 2021*. ACM, 2021.

[13] G. Fuchsbauer, C. Hanser, and D. Slamanig, "Structure-preserving signatures on equivalence classes and constant-size anonymous credentials," *J. Cryptol.*, vol. 32, no. 2, pp. 498–546, 2019.

[14] R. T. Moreno, J. G. Rodríguez, C. T. López, J. B. Bernabé, and A. F. Skarmeta, "OLYMPUS: A distributed privacy-preserving identity management system," in *GIoTS 2020*. IEEE, 2020, pp. 1–6.

[15] E. C. Crites and A. Lysyanskaya, "Delegatable anonymous credentials from mercurial signatures," in *CT-RSA 2019*. Springer, 2019.

[16] J. Blömer and J. Bobolz, "Delegatable attribute-based anonymous credentials from dynamically malleable signatures," in *ACNS 2018*. Springer, 2018.

[17] O. Sanders, "Efficient redactable signature and application to anonymous credentials," in *PKC 2020*. Springer, 2020.

[18] R. T. M. et al, "The OLYMPUS architecture - oblivious identity management for private user-friendly services," *Sensors*, vol. 20, no. 3, p. 945, 2020.

[19] C. Adams, "Achieving non-transferability in credential systems using hidden biometrics," *Security and Communication Networks*, vol. 4, no. 2, pp. 195–206, 2011.

[20] F. Baldimtsi, J. Camenisch, L. Hanzlik, S. Krenn, A. Lehmann, and G. Neven, "Recovering lost device-bound credentials," in *ACNS 2015*. Springer, 2015.

[21] G. L. M. et al., "Technical implementation and feasibility," in *Attribute-based Credentials for Trust: Identity in the Information Society*. Springer, 2015, pp. 255–317.

[22] N. D. Sarier, "Comments on biometric-based non-transferable credentials and their application in blockchain-based identity management," *Computers & Security*, vol. 105, p. 102243, 2021.

[23] M. Blanton and W. M. Hudelson, "Biometric-based non-transferable anonymous credentials," in *ICICS 2009*. Springer, 2009.

[24] R. Impagliazzo and S. M. More, "Anonymous credentials with biometrically-enforced non-transferability," in *WPES 2003*, 2003.

[25] D. Bissessar, C. Adams, and D. Liu, "Using biometric key commitments to prevent unauthorized lending of cryptographic credentials," in *PST 2014*, 2014.

[26] Y. Dodis, L. Reyzin, and A. D. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *EUROCRYPT 2004*. Springer, 2004.

[27] Y. Wen and S. Liu, "Robustly reusable fuzzy extractor from standard assumptions," in *ASIACRYPT 2018*. Springer, 2018.

[28] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. D. Smith, "Reusable fuzzy extractors for low-entropy distributions," *J. Cryptol.*, vol. 34, no. 1, p. 2, 2021.

[29] Q. A. et al, "Pseudoentropic isometries: A new framework for fuzzy extractor reusability," in *AsiaCCS 2018*, 2018.

[30] J. H. Cheon, J. Jeong, D. Kim, and J. Lee, "A reusable fuzzy extractor with practical storage size: Modifying canetti et al.'s construction," in *ACISP 2018*, ser. LNCS, W. Susilo and G. Yang, Eds., vol. 10946. Springer, 2018, pp. 28–44.

[31] K. Zhang, H. Cui, and Y. Yu, "Facial template protection via lattice-based fuzzy extractors," *IACR Cryptol. ePrint Arch.*, p. 1559, 2021.

[32] A. Arakala, J. Jeffers, and K. J. Horadam, "Fuzzy extractors for minutiae-based fingerprint authentication," in *ICB 2007*, ser. LNCS, S. Lee and S. Z. Li, Eds., vol. 4642. Springer, 2007, pp. 760–769. [Online]. Available: https://doi.org/10.1007/978-3-540-74549-5_80

[33] A. Ibarrondo, H. Chabanne, and M. Önen, "Practical privacy-preserving face identification based on function-hiding functional encryption," in *CANS 2021*, ser. LNCS, M. Conti, M. Stevens, and S. Krenn, Eds., vol. 13099. Springer, 2021, pp. 63–71.

[34] J. Lee, D. Kim, D. Kim, Y. Song, J. Shin, and J. H. Cheon, "Instant privacy-preserving biometric authentication for hamming distance," *IACR Cryptol. ePrint Arch.*, p. 1214, 2018.

[35] J. Camenisch, T. Gross, and T. Heydt-Benjamin, "Cryptographic proofs in data processing systems," U.S. Patent 8527777, Sep. 2013.

[36] J. Hesse, N. Singh, and A. Sorniotti, "How to bind anonymous credentials to humans," Cryptology ePrint Archive, Paper 2023/853, 2023, https://eprint.iacr.org/2023/853. [Online]. Available: https://eprint.iacr.org/2023/853

[37] S. Krenn, J. Orlicky, D. Slamanig, and T. Trpišovský, "RiBAC: Strengthening access control systems for pandemic risk reduction while preserving privacy," in *SECPID@ARES 2023*, 2023, (to appear).

[38] N. N. Diep, L. X. Hung, Y. Zhung, S. Lee, Y. Lee, and H. Lee, "Enforcing access control using risk assessment," in *ECUMN 2007*, 2007.

[39] P. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger, "Fuzzy multi-level security: An experiment on quantified risk-adaptive access control," in *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*. IEEE Computer Society, 2007, pp. 222–230. [Online]. Available: https://doi.org/10.1109/SP.2007.21

[40] M. Kowalewski, F. Herbert, T. Schnitzler, and M. Dürmuth, "Proof-of-vax: Studying user preferences and perception of covid vaccination certificates," *Proc. Priv. Enhancing Technol.*, vol. 2022, no. 1, pp. 317–338, 2022.

[41] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[42] A. Ouamane, M. Bengherabi, A. Hadid, and M. Cheriet, "Side-information based exponential discriminant analysis for face verification in the wild," in *IEEE FG*. IEEE, 2015.

[43] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988.

[44] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *CRYPTO 1991*. Springer, 1991, pp. 129–140.

[45] G. Barwell, D. Page, and M. Stam, "Rogue decryption failures: Reconciling AE robustness notions," in *Cryptography and Coding*. Springer, 2015.

[46] J. Groth, "Simulation-sound NIZK proofs for a practical language and constant size group signatures," in *ASIACRYPT 2006*. Springer, 2006.

[47] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *CRYPTO 1997*. Springer, 1997, pp. 410–424.

[48] K. Rannenberg, J. Camenisch, and A. Sabouri, Eds., *Attribute-based Credentials for Trust: Identity in the Information Society*. Springer, 2015.

[49] D. Pointcheval and O. Sanders, "Reassessing security of randomizable signatures," in *CT-RSA 2018*. Springer, 2018.

[50] M. J. Dworkin, *Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac*. National Institute of Standards & Technology, 2007.

[51] H. Li and G. Hua, "Hierarchical-pep model for real-world face recognition," in *IEEE CVPR*, 2015.

[52] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *CRYPTO' 86*, 1987.

[53] S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi, "On the non-malleability of the fiat-shamir transform," in *INDOCRYPT 2012*. Springer, 2012, pp. 60–79.

[54] D. Bernhard, O. Pereira, and B. Warinschi, "How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios," in *ASIACRYPT 2012*. Springer, 2012.

[55] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, and J. Groth, "Foundations of fully dynamic group signatures," in *ACNS 2016*. Springer, 2016.

[56] S. F. Ali, M. A. Khan, and A. S. Aslam, "Fingerprint matching, spoof and liveness detection: classification and literature review," *Frontiers of Computer Science*, vol. 15, no. 1, pp. 1–18, 2021.

[57] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Syst. J.*, vol. 40, no. 3, pp. 614–634, 2001.

Appendix

A. Unforgeability Proof BioABC-R

Proof. In this proof, we will work with a modified version of the *unforgeability* game **ModExp** where the oracle $\mathsf{O}_{present}$ returns a simulated NIZK proof instead of running the $\mathsf{U.Present}$ algorithm. As the protocol Π is zero-knowledge, the two versions of the oracle are computationally indistinguishable, and the experiment modification will result in at most a negligible difference for the winning chances of an adversary.

Let $\mathcal{A}$ be a PPT adversary, we want to prove that $\Pr[\mathcal{A}\text{wins}] = \Pr\left[\mathbf{Exp}_{\mathcal{A}}^{Unforgeability}(1^\lambda) = 1\right] \leq \mathsf{negl}(\lambda)$. Because of the previous discussion, we know that $\Pr[\mathcal{A}\text{wins}] \leq \Pr[\mathcal{A}\text{wins } \mathbf{ModExp}] + \mathsf{negl}(\lambda)$.

To show that the latter is negligible, we construct an adversary $\mathcal{B}$ against Σ's EUF-CMA security in the following way:

- $\mathcal{B}$ receives (pp, pk) as parameters and access to an oracle O_{sign}.
- $\mathcal{B}$ runs $(pt^*, \phi^*, ro_U^*, ctx^*) \leftarrow\$ \mathcal{A}^{\mathsf{O}_{issue}, \mathsf{O}_{present}, \mathsf{O}_{reveal}}(pp, \mathsf{pk})$, answering the oracles as follow:
  - $\mathsf{O}_{issue}(j, Bio_j, \boldsymbol{a}_j)$: $add(\{j, Bio_j, \boldsymbol{a}_j\})$ to Q_{issue}
  - $\mathsf{O}_{present}(j, \phi, ro_U, \mathsf{sk}_{ae}, ctx)$ Simulate the corresponding NIZK proof, returning $\perp$ if the statements are not fulfilled.
  - $\mathsf{O}_{reveal}(j)$: Set $\boldsymbol{b} = (Bio_j, \boldsymbol{a}_j)$. Return the result $\sigma_j \leftarrow\$ \mathsf{O}_{sign}(b)$. Successive calls for j will return the same σ_j.
- If adversary $\mathcal{A}$ did not win, abort.
- If $\mathcal{B}$ fails to extract a witness from pt^*, abort.
- Otherwise, $\mathcal{B}$ extracts a witness $(\sigma^*, Bio^*, \boldsymbol{a}^*)$, sets $\boldsymbol{b} = (Bio^*, \boldsymbol{a}^*)$ outputs $(\sigma^*, \boldsymbol{b})$ as a forgery.

Note that, for $\mathcal{A}$ to win, it cannot have called O_{reveal} for a credential that contains Bio^* (in fact, any Bio that fulfills the matching condition with Bio_f^*), $\boldsymbol{a}^*$ that fulfills the statements (and thus is a valid witness). Further, the winning condition rules out having used one of the presentation tokens received from $\mathsf{O}_{present}$: the exact same token is removed when checking the contents of $Q_{present}$, and trying to use a different fresh biometric for the forgery will be detected by $\mathsf{R.GenEphVerifier}$. Indeed, the adversary controls ri_V, which should be the encrypted biometric template of the user in the credential in an honest flow plus the commitment used for the zero-knowledge proof. If the adversary modifies ri_V to an encryption of a different value (so it matches the fresh biometric value), then opening check in the device will fail with overwhelming probability (as the commitment scheme is computationally binding). However, if the adversary tries to modify the commitment itself, then the zero-knowledge proof will fail with overwhelming probability (again, because of the computationally binding property), as it checks that the opening matches the value in the credential.

Therefore, $(\sigma^*, \boldsymbol{b})$ is an actual forgery in Σ's EUF-CMA experiment, as it was not received from $\mathsf{O}_{sign}(b)$. Further, the way $\mathcal{B}$ answers the oracle queries is consistent with how the **ModExp** game does.

Taking those arguments into account, $\Pr[\mathcal{A}\text{wins } \mathbf{ModExp}] = \Pr[\mathcal{A}\text{wins } \mathbf{ModExp} \wedge \text{extfails}] + \Pr[\mathcal{A}\text{wins } \mathbf{ModExp} \wedge \neg\text{extfails}]$, while $\Pr[\mathcal{A}\text{wins } \mathbf{ModExp} \wedge \neg\text{extfails}] \leq \Pr[\mathcal{B}\text{wins}]$. The probability of a failed witness extraction of a forged token is negligible due to Π being simulation-sound extractable, and the adversary winning without outputting a real forged token is negligible (as discussed above). What is more, the probability of $\mathcal{B}$ winning is also negligible, as Σ is EUF-CMA-secure. With this, the proof is finished. □

B. Unlinkability Proof of BioABC-R

Proof. The proof involves a series of modifications of the unlinkability experiment supported by the construction's building block's security properties.

Note that in the winning conditions, the witnesses are ensured to be valid, i.e., the credentials are valid and the predicates for the attributes are fulfilled in both cases. If the values generated by the adversary do not correctly meet these criteria, the output is a random bit, giving no advantage to the adversary. Thus, we can modify the experiment to have the challenger return a simulated proof instead of running $\mathsf{U.Present}$, and the result will be computationally indistinguishable for the adversary due to Π being zero-knowledge.

In $\mathsf{R.GenEphVerifier}$, ri_V^* is expected to be an encrypted value and a commitment, for which the adversary does not know any information about the secret keys, nor the committed value. If authentication of the ciphertext fails, the device will always abort. Therefore, unless the adversary can forge an authenticated encryption (which has negligible probability), it cannot modify the encrypted value. Furthermore, if the adversary modifies the committed value, the opening check will fail (except for at most negligible probability) because of the hiding property of the commitments, and the result will always be 0, giving no advantage. Thus, we can modify the game to remove the adversary's control over ri_V with a negligible change in advantage. That is, the adversary still receives the honestly generated $ri_V = \{ae_{a_{Bio}}, C_{a_{Bio}}\}$ from $pt*$, but the adversary's output ri_V* is ignored. Instead, the challenger honestly generates ri_V using the $V.InputGen$ protocol, and the value is subsequently used in $\mathsf{R.GenEphVerifier}(Bio_f^b, ri_V)$.

What is more, we can further modify the experiment so the adversary does not even receive the honest value of ri_V, but random elements instead, with a negligible change in advantage. This is due to the indistinguishability of the encryption scheme $\mathcal{E}$, and the hiding property of the commitment scheme.

Up until this point, we have obtained a modified experiment **ModExp** in which the adversary only gets a simulated token, the ro_U^* value (which, in this construction, is null), and the ro_V^* returned by the device. The first two values are clearly independent of b, and the third is also independent of b because of the restriction that $\mathcal{M}(Bio_f^t, a_{Bio}{}^t) = 1\ t \in \{0,1\}$, as in this game it consists on a honest verification of the matching between the templates, which will return 1 regardless of the value of b. Thus, $\Pr[\mathbf{ModExp}_{\mathcal{A}}] = \frac{1}{2}$. As we have proved in the description of each modification, the difference of advantages in each step is negligible, so the winning chance between the original experiment and **ModExp** differs in at most a negligible quantity. That is, $\Pr\left[\mathbf{Exp}_{\mathcal{A}}^{Unlinkability}(1^\lambda) = 1\right] \leq \Pr[\mathbf{ModExp}_{\mathcal{A}}] + \mathsf{negl}(\lambda)$, and this concludes the proof. □

C. Unforgeability Proof BioABC-ZK

Proof. In this proof, we will work with a modified version of the *unforgeability* game **ModExp** where the oracle $\mathsf{O}_{present}$ returns a simulated NIZK proof instead of running the U.Present algorithm. As the protocol Π is zero-knowledge, the two versions of the oracle are computationally indistinguishable, and this modification will result in at most a negligible difference for the winning chances of an adversary.

Let $\mathcal{A}$ be a PPT adversary, we want to prove that $\Pr[\mathcal{A}\text{wins}] = \Pr\left[\mathbf{Exp}_{\mathcal{A}}^{Unforgeability}(1^\lambda) = 1\right] \leq \mathsf{negl}(\lambda)$. Because of the previous discussion, we know that $\Pr[\mathcal{A}\text{wins}] \leq \Pr[\mathcal{A}\text{wins } \mathbf{ModExp}] + \mathsf{negl}(\lambda)$.

To show that the latter is negligible, we construct an adversary $\mathcal{B}$ against Σ's EUF-CMA security in the following way:

- $\mathcal{B}$ receives (pp', pk) as parameters and access to an oracle O_{sign}.
- $\mathcal{B}$ runs $(pt^*, \phi^*, Bio_f^*, ri_V^*, ctx^*) \leftarrow\$ \mathcal{A}^{\mathsf{O}_{issue}, \mathsf{O}_{present}, \mathsf{O}_{reveal}}(pp', \mathsf{pk})$ answering the oracles as follows:
  - $\mathsf{O}_{issue}(j, a_{Bioj}, \boldsymbol{a}_j)$: $add(\{j, a_{Bioj}, \boldsymbol{a}_j\})$ to Q_{issue}
  - $\mathsf{O}_{present}(j, \phi, ro_U, \mathsf{sk}_{ae}, ctx)$ Simulate the corresponding NIZK proof, returning $\perp$ if the statements are not fulfilled.
  - $\mathsf{O}_{reveal}(j)$: Set $\boldsymbol{b} = (a_{Bioj}, \boldsymbol{a}_j)$. Return $\sigma_j \leftarrow\$ \mathsf{O}_{sign}(b)$. Successive calls for j will return the same σ_j.
- If adversary $\mathcal{A}$ did not win, abort.
- If $\mathcal{B}$ fails to extract a witness from pt^*, abort.
- Otherwise, $\mathcal{B}$ extracts a witness $(\sigma^*, a_{Bio}{}^*, \boldsymbol{a}^*)$, sets $\boldsymbol{b} = (a_{Bio}{}^*, \boldsymbol{a}^*)$ outputs $(\sigma^*, \boldsymbol{b})$ as a forgery.

Note that, for $\mathcal{A}$ to win, it cannot have called O_{reveal} for a credential that contains $a_{Bio}{}^*$ (in fact, any a_{Bio} that fulfils the matching condition with Bio_f^*), and $\boldsymbol{a}^*$ that fulfils the statements. Further, the winning condition rules out having used one of the presentation tokens received from $\mathsf{O}_{present}$: the exact same token is removed when checking the contents of $Q_{present}$, and trying to use a different fresh biometric for the forgery will be detected by R.GenEphVerifier. In particular, if the commitment to the fresh biometric template is modified, it will be detected as $verfFP$ returns an honest value. Otherwise, changing the underlying fresh biometric values would require breaking the commitment scheme, which has negligible probability as the commitment scheme is computationally binding.

Therefore, $(\sigma^*, \boldsymbol{b})$ is an actual forgery in Σ's EUF-CMA experiment, as it was not received from $\mathsf{O}_{sign}(b)$. Further, the way $\mathcal{B}$ answers the oracle queries is consistent with how the **ModExp** game does.

Taking those arguments into account, $\Pr[\mathcal{A}\text{wins } \mathbf{ModExp}] = \Pr[\mathcal{A}\text{wins } \mathbf{ModExp} \land \text{extfails}] + \Pr[\mathcal{A}\text{wins } \mathbf{ModExp} \land \neg\text{extfails}]$, while $\Pr[\mathcal{A}\text{wins } \mathbf{ModExp} \land \neg\text{extfails}] \leq \Pr[\mathcal{B}\text{wins}]$. The probability of a failed witness extraction of a forged token is negligible due to Π being simulation-sound extractable, and the adversary winning without outputting a real forged token is negligible (as discussed above). What is more, the probability of $\mathcal{B}$ winning is also negligible, as Σ is EUF-CMA-secure. This concludes the proof. □

D. Unlinkability Proof of BioABC-ZK

Proof. We perform a series of modifications of the unlinkability experiment supported by the building block's security properties, as well as the construction's procedures. First we observe that in the winning conditions, the witnesses are ensured to be valid, i.e., the credentials are valid and the predicates and matching condition for biometric data are fulfilled in both cases. If this is not fulfilled, the output is a random bit, giving no advantage to the adversary. Thus, we can modify the experiment to have the challenger return a simulated proof instead of running U.Present, and the result will be computationally indistinguishable for the adversary due to Π being zero-knowledge.

The verifier is in charge of forwarding the result of the algorithm R.GenEphUser executed by the reader device, so the experiment models the possibility of the adversary modifying the ro_U^* value. However, the construction involves authenticated encryption, achieving non-malleability. Thus, if the adversary tries to modify the value, the user will abort unless the adversary was able to forge an authenticated encryption, which has negligible probability for an authenticated encryption scheme $\mathcal{E}$. Therefore, we can modify the game to remove the adversary's control over ro_U^*, that is, to let the challenger receive directly the result coming from $\mathsf{R.GenEphUser}$ for its computations, and the difference in advantage for the for the adversary will be negligible.

In this construction, the value of ri_V^* is ignored by algorithm $\mathsf{R.GenEphVerifier}$, so the challenger can perform the ro_V^* generation before the simulated proof, and the change is completely transparent to the adversary. With this technical modification, the challenger gets the public commitment used in the zero-knowledge proof without needing the public part coming from the ro_U^* value.

The ro_U^* value, even if not needed by the challenger, and not controlled by the adversary, is still an input for the adversary that depends on the chosen bit (the fresh fingerprint, specifically). In this construction ro_U^* is a ciphertext encrypted with a fresh key. As the adversary does not control the corresponding secret key, and $\mathcal{E}$ is IND-CPA-secure, we can substitute the honest ro_U^* for a randomly chosen encrypted value and the result will be computationally indistinguishable to the adversary.

Lastly, ro_V^* is a commitment to the fresh biometric template. However, as $\mathcal{C}$ is a computationally hiding commitment scheme, the value can be substituted by a commitment to any other random value.

After these modifications, the resulting experiment **ModExp** does not give the adversary $\mathcal{A}$ any input related to the chosen bit b. Thus, $\Pr[\mathbf{ModExp}_{\mathcal{A}}] = \frac{1}{2}$. Because each modification to the experiment is computationally indistinguishable from the previous one to the adversary, the difference of advantages between the original experiment and **ModExp** is at most negligible, that is, $\Pr\left[\mathbf{Exp}_{\mathcal{A}}^{Unlinkability}(1^\lambda) = 1\right] \leq \Pr[\mathbf{ModExp}_{\mathcal{A}}] + \mathsf{negl}(\lambda)$, which concludes the proof. $\square$