AirBB: Lightweight Threshold Anonymous Credentials without Random Oracles from BB Signatures

Sebastian Faust, Jonas Hofmann, Philipp-Florens Lehwalder, Hussien Othman
Technical University of Darmstadt
{sebastian.faust,jonas.hofmann1,philipp-florens.lehwalder}@tu-darmstadt.de
hussien.othman@gmail.com

November 6, 2026

Abstract

Anonymous credentials are a well-studied cryptographic primitive enabling users to prove statements about their identity and selectively present personal information in an unlinkable manner. The issuer, however, is often a single point of failure. Threshold issuance addresses this by distributing the issuer across several parties such that a quorum of them must cooperate to issue credentials. Numerous constructions of single-issuer and threshold issuance anonymous credentials have been proposed. However, existing schemes each have at least one of the following weaknesses: (1) their presentations depend on the number of attributes, (2) they are not concretely efficient, or (3) they rely on the random oracle model (ROM) for non-interactive presentations.

In this work, we present a new construction of *threshold* anonymous credentials that has a constant presentation size of 240 bytes, rivals the efficiency of BBS, and does not rely on the random oracle model. We achieve this by combining techniques from set commitments with Boneh-Boyen (BB) signatures (Eurocrypt'04) and prove our scheme's security in the algebraic group model. This is on par with the tight security proofs of existing schemes, which typically rely on the AGM or GGM, or on assumptions justified in those models. Since signing uses linear operations, the scheme is easily thresholdizable, without requiring interaction between the issuers or preprocessing.

1 Introduction

Anonymous credentials, as envisioned by Chaum [22] in 1985 and first formally proposed by Camenisch and Lysyanskaya [16, 17, 18], allow users to obtain credentials certifying a collection of attributes and to prove statements about attribute subsets. Crucially, presentations of the same credential must remain unlinkable to one another. An anonymous credential (AC) scheme typically consists of two phases. The *issuance phase* is an interactive protocol between a holder and an issuer, in which the holder, possessing a set of attributes M, obtains a credential on M from the issuer. The *presentation phase* allows the holder to prove possession of a valid credential to a verifier. The two main security requirements of an anonymous credential are: (i) *unforgeability*, guaranteeing that only an authorized issuer can issue valid credentials, (ii) *unlinkability*, meaning that multiple presentations of the same credential cannot be linked to one another or to the issuance session.

Anonymous credentials [22, 16] have seen renewed interest, driven by the prospect of their adoption in future iterations of the EUDI wallet [28, 27, 9], a European initiative to offer every EU citizen a digital identity. Consequently, a large body of work explores different trade-offs between efficiency, security, and functionality. One of the most widely investigated AC schemes is based on BBS signatures [8, 4, 12, 53], which offer concretely efficient issuance, presentation, and verification. Consequently, BBS-based credentials are the subject of ongoing standardization efforts at the IETF [41], the W3C [6], and ISO [1]. While attractive due to their efficiency and simplicity, BBS-based credentials suffer from several drawbacks:

1. *Presentation size:* The size of the presentation proof grows linearly with the number of hidden attributes. For example, a presentation proof with 30 hidden attributes is already a kB in size.

2. *Use of the ROM:* The presentation protocol uses the Fiat-Shamir heuristic [30], and thus relies on the random oracle model, on top of a further idealized model.[1]

3. *Threshold issuance:* Distributing the issuance process has emerged as an attractive solution to avoid single points of failure. While there exist schemes for thresholdizing the issuance, they require executing a costly multi-round interactive MPC protocol among the issuers, which stems from the non-linearity of the BBS signature scheme [26].

While each of these three limitations has been addressed individually in the literature (see Section 1.2 for a comparison), no existing scheme overcomes all of them simultaneously while remaining concretely efficient. In this work, we present a new anonymous credential scheme that addresses all three limitations at once, with efficiency on par with BBS-based credentials.

1.1 Our contribution

The main contribution of this work is an anonymous credential system based on the Boneh-Boyen (BB) signature scheme [7]. Despite its attractive features, in particular, short signatures and efficient signing, BB signatures have, somewhat surprisingly, not been used as the basis for anonymous credentials (we discuss one very recent exception in Section 1.2). Our construction achieves credential sizes as well as issuance and presentation run-times competitive with those of BBS-based systems, while overcoming the three shortcomings described above. We discuss our contributions in more detail below.

1. We present an efficient AC construction based on BB signatures. To the best of our knowledge, our construction is the *first* credential scheme that does not require the ROM while having similar efficiency as BBS. Further, it achieves the smallest presentation size among credential schemes with constant-size presentations to date.

2. Our scheme allows for efficient thresholdization. To the best of our knowledge, this is the *first* practical, non-interactive threshold construction with constant-size presentations that does not require the random oracle model. The only other construction achieving constant-size presentations without the ROM and without interaction [35] has presentations that are more than an order of magnitude larger (5184 versus 240 bytes). We further prove our scheme secure against an adversary making adaptive corruptions.

3. We implement our construction in Rust and demonstrate its practicality, showing that its runtime for issuance is under 2.1ms, presentation and verification times are on par with BBS signatures, and presentations are only 240 bytes in size.

1.2 State of the art

Below, we give an overview of existing (threshold) anonymous credential schemes and summarize how they compare to our construction, including their respective shortcomings, in Table 1.

BBS-based anonymous credentials. Many state-of-the-art anonymous credential schemes build on the BBS signature family [8, 4, 12, 53]. These constructions are concretely efficient (short signature sizes and an efficient issuance protocol), but all of them suffer from at least one of the drawbacks mentioned.

First, distributing the issuance process among several issuers faces inherent challenges stemming from the non-linearity of the BBS signing operation, which necessitates either multiple rounds of online interaction or a preprocessing phase [26, 37, 29]. Both are costly, in particular when the number of issuers (or signatures) is large. Second, in the presentation protocol, the holder proves possession of a credential via a non-interactive zero-knowledge proof (NIZK). Such NIZKs are typically obtained through the Fiat–Shamir transform [30], which requires the random oracle model as an additional idealized assumption and, since the resulting proofs involve a hash function evaluation, makes features such as aggregation and re-randomization difficult to achieve. Third, the size of the presentation proof grows linearly with the number of hidden attributes. Tan and Groß [52] recently gave a construction of BBS+ credentials with constant-size presentation proofs, but their construction still suffers from the other two drawbacks.

[1] BBS requires a security proof in the AGM when aiming for a tight reduction [53], unless only distinct messages are signed [20].

Structure-preserving signatures. Another line of work builds anonymous credentials from structure-preserving signatures on equivalence classes [32, 23, 36, 13] combined with randomizable set commitments. In these schemes, commitments to attributes are treated as representatives of equivalence classes, enabling the credential signature to be publicly randomized between presentations without invalidating it. Using randomizable set commitments for the attributes, this approach yields constant-size subset openings and constant-size presentation proofs, without zero-knowledge proofs and without the ROM. However, credentials built from structure-preserving signatures usually have larger signatures and presentations and require more pairing evaluations during verification, which is why deployment efforts [25, 45, 14, 10, 11] have mostly focused on the BBS-based constructions discussed above.

In addition, most such schemes still require an interactive presentation protocol, in which the verifier provides a fresh challenge, or rely on the random oracle model to ensure the freshness of presentation proofs. Several works consider threshold structure-preserving signatures [35, 24, 43], which often benefit from aggregatability, i.e., partial signatures from different issuers can be combined non-interactively into a single signature, and from a non-interactive signing process; however, these schemes either require the random oracle model, have linear-sized presentations or lack concrete efficiency.

Credentials from zero-knowledge proof systems. A recent line of work considers anonymous credentials built from legacy signatures and zkSNARK proof systems [44, 31, 39, 48, 47]. CRS-based zkSNARKs, such as Groth16, also do not rely on the random oracle model. While their proof size and verification time rival that of dedicated anonymous credential schemes, their proving time is far greater, usually in the order of hundreds of milliseconds. Transparent NARKs, such as Ligero [3], have better prover efficiency and do not require a CRS, but usually use Fiat-Shamir to be rendered non-interactive. A special case of proof-system-based credentials is the single-attribute credential scheme of Belenkiy et al. [5]. It achieves non-interactive presentation proofs without the ROM by relying on Groth-Sahai proofs, but is not concretely efficient due to the large number of pairings involved, and it does not readily extend to support multiple attributes.

PS-based credentials. Another important line of work focuses on building anonymous credentials from the PS signature scheme [46]. Compared to BBS, PS signatures are larger, and the key sizes depend on the maximum number of signable attributes. Recently, Sanders [49] showed how to build constant-size credentials from PS signatures. In comparison to our work, that construction is slightly larger and less efficient, while also relying on the random oracle model.

On the positive side, PS signatures are aggregatable and admit a straightforward thresholdization [51].

Threshold blind BB signatures. Most related to our work is the recent work of Fuchsbauer et al. [34] who proposed a blind signature construction from BB signatures, which is readily thresholdizable and enjoys a highly efficient issuance procedure without the random oracle model. They further give a presentation protocol, making theirs the one exception noted above; it works similarly to that of BBS, however, and requires the random oracle model to achieve non-interactivity. Furthermore, their presentation proof size grows linearly with the number of hidden attributes.

1.3 Roadmap

In Section 2, we present a technical overview of our approach. Next, in Section 3, we introduce the necessary preliminaries. In Section 4, we formalize the notion of (threshold) anonymous credentials. In Section 5, we present our threshold signature scheme BB-KZG, and in Section 6, we give its blind variant Blind-BB-KZG. We then present our credential construction and establish its security properties in Section 7. Finally, in Section 8, we report our benchmark results.

2 Technical Overview

We give an overview of the main ideas behind our construction and its security proof. Our starting point is the blind Boneh–Boyen (BB) signature construction of [34]. BB signatures are particularly attractive for threshold anonymous credentials: signatures are short, the signing operation is linear in the secret key, and threshold signatures can therefore be obtained by independently computing partial signatures and aggregating them using Lagrange coefficients. Moreover, BB signatures do not inherently rely on the random oracle model.

AC Scheme	$\lvert\sigma\rvert$	$\lvert\pi\rvert$	Assumption	ROM
BBS [53]	$1G_1+1F$	$2G_1+(3+\lvert\mathsf{Hid}\rvert)F$	q-SDH†/ DL, AGM	●
MoniPoly [52]	$1G_1+2F$	$2G_1+7F$	q-SDH†/ DL, AGM	●
PS [46]	$2G_1$	$2G_1+(2+\lvert\mathsf{Hid}\rvert)F$	LRSW(GGM)	●
Sanders [49]	$2G_2$	$2G_1+2G_2+2F$	GGM, DDH	●
SPS-EQ [32]	$3G_1+1G_2+2F$	$3G_1+1G_2+5F$	GGM, q-DL, q-GSDH	●
BlindBB [34]	$2G_1$	$2G_1+(2+\lvert\mathsf{Hid}\rvert)F$	DL, AGM	●
Ours	$2G_1$	$3G_1+1G_2$	q-DL, AGM	○

Threshold ACs	$\leftrightarrow$	$\lvert\pi\rvert$	Assumption	ROM
Threshold Mercurial Sig [2]	●	$3G_1+1G_2+5F$	GGM	●
Threshold BBS+ [26, 29]	●	$2G_1+(4+\lvert\mathsf{Hid}\rvert)F$	q-SDH	●
Silent Threshold ACs [35]	○	$44G_1+32G_2$	GGM	○
Coconut [51]	○	$3G_1+1G_2+(2+\lvert\mathsf{Hid}\rvert)F$	LRSW(GGM)	●
Threshold BlindBB [34]	○	$2G_1+(2+\lvert\mathsf{Hid}\rvert)F$	DL, AGM	●
Ours	○	$3G_1+1G_2$	q-DL, AGM	○

Table 1: Comparison of our single-issuer and threshold constructions with related schemes. "ROM" marks whether a scheme relies on the random oracle model, and "$\leftrightarrow$" whether threshold issuance requires interaction among issuers or pre-generated randomness. $\lvert\mathsf{Hid}\rvert$ denotes the number of hidden attributes, G_1 and G_2 refer to an element of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and F refers to a field element. † marks assumptions under which the security proof is not tight; for BBS, a tight standard-model reduction is known when only distinct messages are signed [20].

The main difficulty is selective disclosure. The construction of [34] encodes multiple attributes using Pedersen vector commitments and proves possession of the hidden attributes using Schnorr-type proofs. Consequently, the presentation proof grows with the number of hidden attributes, and making it non-interactive requires Fiat–Shamir.

At a high level, our main idea, following [32], is to replace this vector encoding by a polynomial encoding based on KZG commitments. This gives a single algebraic representation that can be signed using BB, blinded during issuance, and opened with respect to any subset of attributes using a constant-size witness. In the remainder of this section, we discuss how to adapt this idea to BB signatures and extend it to an anonymous credential scheme.

2.1 Our Construction

We next overview the main ideas of the construction.

BB **signatures and thresholdization.** We first recall the structure of the BB signature that is relevant to our construction. Let $\mathsf{sk} \xleftarrow{\$} \mathbb{F}_p$ and $\mathsf{vk} = \mathbf{e}(g_1, g_2)^{\mathsf{sk}}$. A BB-style signature has the form $\sigma = \left(g_1^{\mathsf{sk}} Y^r, g_1^r\right)$ for a message-dependent group element Y and uniformly random $r \xleftarrow{\$} \mathbb{F}_p$. Verification is expressed by a pairing equation relating Y and the two signature components. The important feature for us is that the signing key occurs linearly in the exponent. For a t-out-of-n threshold scheme, we Shamir-share sk as $\{\mathsf{sk}_i\}_{i\in[n]}$. Issuer i computes a partial signature using its share, and any t partial signatures can be aggregated component-wise using Lagrange coefficients. Thus, unlike thresholdizations of signature schemes involving non-linear signing operations, the issuers need not execute an online MPC protocol.

Encoding attribute sets using KZG **commitments.** To sign a set of attributes while retaining a compact algebraic representation, we use the polynomial structure underlying KZG commitments [38].

Given an attribute set M, define its root polynomial

$$p_M(X) := \prod_{m \in M} (X - m).$$

We use a structured common reference string (CRS) containing powers of a secret evaluation point τ. In particular, the CRS contains sufficiently many powers of τ in both source groups to evaluate the relevant polynomials in the exponent without knowing τ. For $h_1 = g_1^{\nu}$ and $h_2 = g_2^{\nu}$, the element $C_M := h_1^{p_M(\tau)}$ is a KZG commitment to p_M, and hence is a compact commitment to the entire attribute set M. The same polynomial can analogously be evaluated in $\mathbb{G}_2$ as $h_2^{p_M(\tau)}$.

This polynomial representation is useful for selective disclosure because set containment becomes polynomial divisibility. If $\mathsf{Rev} \subseteq M$ and $\mathsf{Hid} = M \setminus \mathsf{Rev}$, then $p_M(X) = p_{\mathsf{Rev}}(X)\, p_{\mathsf{Hid}}(X)$. Hence, given p_M and Rev, an evaluation of the quotient polynomial p_{Hid} in the exponent serves as a constant-size witness that every revealed attribute is contained in M.

BB-KZG. We first incorporate this polynomial representation directly into the BB signature, by replacing the message-dependent element Y with the KZG commitment $h_1^{p_M(\tau)}$. In more detail, a signature on the attribute set M is

$$\sigma = (\sigma_1, \sigma_2) = \left(g_1^{\mathsf{sk}} h_1^{r p_M(\tau)}, g_1^{r}\right), \qquad r \xleftarrow{\$} \mathbb{F}_p,$$

and verification checks

$$\mathbf{e}(\sigma_1, g_2) \stackrel{?}{=} \mathsf{vk} \cdot \mathbf{e}\left(\sigma_2, h_2^{p_M(\tau)}\right).$$

This modification preserves the linear structure of BB signatures and hence allows for simple thresholdization.

Intuitively, security is preserved because the randomness r binds the two signature components to the same polynomial evaluation $p_M(\tau)$. Given a signature on a polynomial p_i, an adversary sees $\left(g_1^{\mathsf{sk}} h_1^{r_i p_i(\tau)}, g_1^{r_i}\right)$. Although the adversary may choose fresh randomness $r^\star$ and can compute the corresponding terms $g_1^{r^\star}$ and $h_1^{r^\star p^\star(\tau)}$ for a target polynomial $p^\star$, it still needs the signing-key contribution g_1^{sk}. Previous signatures expose this contribution only masked by terms of the form $h_1^{r_i p_i(\tau)}$, where the signing randomness r_i is unknown. Thus, forging a signature on a fresh polynomial requires algebraically removing or recombining these masks to obtain the signing-key contribution.

Note that BB signatures are originally proven secure only against a selective adversary, who chooses the message to be forged in advance. In our setting, however, we want adaptive security, i.e., the adversary may choose the forged attribute set only after observing signatures on adaptively chosen sets.

We follow the approach of [34], who showed a proof for blind BB signatures in the algebraic group model (AGM). In the AGM, every group element output by the adversary is accompanied by an algebraic representation in terms of previously received group elements. Substituting these representations into the verification equation yields a polynomial relation in the secret setup values, signing keys, and signature randomness. We discuss our proofs in more detail in Section 2.2.

We next discuss how to use this signature scheme in constructing anonymous credentials. We first transform BB-KZG into a partially blind signature scheme Blind-BB-KZG. Then, we combine the scheme with a presentation protocol.

Blind-BB-KZG. Suppose the holder wants a signature on M, while revealing only $\mathsf{Rev} \subseteq M$ to issuer i. Let $\mathsf{Hid} := M \setminus \mathsf{Rev}$, so that $p_M = p_{\mathsf{Rev}} \cdot p_{\mathsf{Hid}}$. The holder samples $a \xleftarrow{\$} \mathbb{F}_p^*$ and computes $C := h_1^{a p_M(\tau)}$. This is a multiplicatively blinded KZG commitment to the attribute polynomial: multiplication by a preserves its roots, while hiding the polynomial, and hence the hidden attributes, from the issuer.

The issuer signs C as before. After verifying the partial signature, the holder unblinds by setting

$$\sigma_1 := \sigma_1', \qquad \sigma_2 := (\sigma_2')^a.$$

Writing $r' := ar$, the resulting signature is

$$\left(g_1^{\mathsf{sk}_i} h_1^{r' p_M(\tau)}, g_1^{r'}\right),$$

and hence has the same distribution as an ordinary BB-KZG partial signature.

There is, however, an important subtlety: the issuer cannot safely sign an arbitrary group element C. For example, a malicious holder can choose $x \in \mathbb{F}_p^*$ and send $C := g_1^x$ instead of a properly formed commitment. The issuer then computes

$$\sigma_1' = g_1^{\mathsf{sk}_i} C^r = g_1^{\mathsf{sk}_i + xr}, \qquad \sigma_2' = g_1^r.$$

Since the holder knows x, it can remove the randomization term:

$$\frac{\sigma_1'}{(\sigma_2')^x} = \frac{g_1^{\mathsf{sk}_i + xr}}{g_1^{xr}} = g_1^{\mathsf{sk}_i}.$$

Thus, a single malicious issuance query reveals the group element $g_1^{\mathsf{sk}_i}$ corresponding to the issuer's signing share, which can then be used to construct arbitrary partial signatures for that issuer. We therefore need to ensure that C is a well-formed blinded KZG commitment. The attack above succeeds because the adversary knows an explicit scalar x such that $C = g_1^x$, and can consequently derive $C^r = (g_1^r)^x$ from the issuer's second response. For a well-formed commitment $C = h_1^{p(\tau)}$, no such known scalar relation is available: although the adversary receives g_1^r, it does not know r or the secret setup value τ, and hence cannot analogously recover the factor $C^r = h_1^{rp(\tau)}$.

To this end, to enforce well-formedness, the holder additionally computes $\hat{C} := g_2^{ap_{\mathsf{Hid}}(\tau)}$, which acts as a KZG-style divisibility witness. The issuer verifies

$$e(C, g_2) \stackrel{?}{=} e\left(h_1^{p_{\mathsf{Rev}}(\tau)}, \hat{C}\right). \tag{1}$$

For an honestly generated query, it holds that $ap_M = p_{\mathsf{Rev}} \cdot (ap_{\mathsf{Hid}})$, so the verification succeeds. Intuitively, the check ensures that the polynomial encoded in C is divisible by p_{Rev}; equivalently, all revealed attributes are roots of that polynomial.

The remaining question is whether a malicious holder can satisfy the pairing check using group elements that do not admit such a polynomial representation. We rule this out in the AGM. In particular, we prove that from every accepting pair $(C, \hat{C})$ satisfying Equation (1), a straight-line extractor can recover a polynomial p such that $C = h_1^{p(\tau)}$ and $p_{\mathsf{Rev}} \mid p$.

We note that one particular challenge in proving the argument above stems from the need to handle adaptive corruptions in the threshold setting. To address this, we place C and $\hat{C}$ in different source groups. We explain the reason behind this in the proof overview in Section 2.2. We stress that this distinction is specific to the threshold setting. For $t = 1$, both elements could instead be placed in the same source group.

From Blind-BB-KZG to anonymous credentials. We finally add a presentation protocol to our blind signature scheme Blind-BB-KZG. A presentation must show that the holder possesses a valid signature on some M containing the revealed subset Rev, while hiding $\mathsf{Hid} = M \setminus \mathsf{Rev}$. At the same time, presentations of the same credential must be unlinkable.

Rerandomization. A first challenge is rerandomization. Additive randomization of the polynomial encoding would change its roots and therefore change the credential's attributes. Conversely, applying the same multiplicative mask to all components of the signature and the commitment does not preserve the BB-KZG verification equation. We therefore combine the two forms of randomization. Let $\sigma = (\sigma_1, \sigma_2)$ be a credential on M. The holder samples fresh $a_0 \xleftarrow{\$} \mathbb{F}_p, a_1 \xleftarrow{\$} \mathbb{F}_p^*$ and computes

$$\sigma_1' := \sigma_1 h_1^{a_0 p_M(\tau)}, \qquad \sigma_2' := (\sigma_2 g_1^{a_0})^{1/a_1},$$

together with

$$C := h_2^{a_1 p_M(\tau)}, \qquad \hat{C} := g_1^{a_1 p_{\mathsf{Hid}}(\tau)}.$$

The presentation is verified using:

$$\mathbf{e}(g_1, C) \stackrel{?}{=} \mathbf{e}\left(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}\right)$$

and the BB-KZG signature equation

$$\mathbf{e}(\sigma_1', g_2) \stackrel{?}{=} \mathsf{vk} \cdot \mathbf{e}(\sigma_2', C).$$

The first equation establishes that the polynomial represented by C contains all revealed attributes, while the second establishes that this is a valid signature on the same polynomial. Note that the presentation contains only a constant number of group elements, independent of the number of hidden attributes.

Unlinkability. The two randomizers are chosen jointly to preserve both verification equations. The additive randomizer a_0 refreshes the BB signing randomness, while a_1 multiplicatively rerandomizes the polynomial encoding.

In fact, this gives information-theoretic unlinkability. Consider two credentials on attribute sets M_0 and M_1 that have the same revealed attributes. For any randomness (a_0, a_1) used to present the first credential, there is a unique pair (a_0', a_1') producing exactly the same presentation from the second credential. This defines a bijection between the two randomness spaces, and hence the resulting presentation distributions are identical even against an unbounded adversary.

Replay Attacks. Rerandomizability creates a second difficulty. An adversary that observes a valid presentation could itself rerandomize the presentation and submit the result as its own. In a NIZK-based construction, this is not an issue because rerandomizing the credential would also require generating a fresh proof. Our presentation, on the other hand, consists only of algebraic group elements, so we must rule out this attack explicitly.

We bind every presentation to auxiliary data aux, which may, for example, contain a verifier identity together with a nonce or timestamp. A natural attempt would be to append aux to the revealed attributes. This is not possible because the underlying signature authenticates a fixed attribute set chosen at issuance. Instead, we incorporate aux only into the KZG witness, through the factor $X - \mathsf{aux}$, without modifying the signed polynomial. Concretely, the holder now computes

$$C := h_2^{a_1 p_M(\tau)}, \qquad \hat{C} := g_1^{a_1 p_{\mathsf{Hid}}(\tau)(\tau - \mathsf{aux})},$$

and the opening equation is replaced by

$$\mathbf{e}\left(g_1^{\tau - \mathsf{aux}}, C\right) \stackrel{?}{=} \mathbf{e}\left(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}\right). \tag{2}$$

The randomized signature components remain unchanged, and are still verified using

$$\mathbf{e}(\sigma_1', g_2) \stackrel{?}{=} \mathsf{vk} \cdot \mathbf{e}(\sigma_2', C). \tag{3}$$

Thus, the presentation is now bound to aux, while the credential itself remains a signature only on M. In particular, rerandomizing an observed presentation does not yield a valid presentation for fresh auxiliary data. To see the intuition, suppose the adversary observes a presentation containing $\hat{C} = g_1^{a_1 p_{\mathsf{Hid}}(\tau)(\tau - \mathsf{aux})}$. To reuse it for some $\mathsf{aux}' \neq \mathsf{aux}$, it would need to produce $\hat{C}' = g_1^{a_1 p_{\mathsf{Hid}}(\tau)(\tau - \mathsf{aux}')}$, so that

$$e\Big(g_1^{\tau - \mathsf{aux}'}, C\Big) = e\Big(\hat{C}', h_2^{p_{\mathsf{Rev}}(\tau)}\Big)$$

holds. In other words, the adversary would have to remove the factor $(\tau - \mathsf{aux})$ from the exponent of the old witness and replace it by $(\tau - \mathsf{aux}')$. Equivalently, it would need to raise $\hat{C}$ to $\frac{\tau - \mathsf{aux}'}{\tau - \mathsf{aux}}$. Since τ is the secret setup value, this transformation is infeasible for the adversary.

We formalize this intuition in the AGM. In the polynomial relation induced by Equation (2), terms stemming from an earlier presentation with auxiliary value aux_j can cancel only if the new auxiliary value satisfies $\mathsf{aux} = \mathsf{aux}_j$. Hence, previous presentation responses cannot be algebraically reused.

Freshness. Freshness ensures that a presentation was generated only after a fresh challenge was issued, and hence was not prepared arbitrarily far in advance. This is different from context binding, which only ensures that a presentation is tied to a particular context, but does not guarantee when it was generated. Our construction can provide freshness by letting the verifier choose aux unpredictably, for example as a fresh random nonce together with the verifier identity and session context. Since a presentation bound to one auxiliary value cannot be transformed into a valid presentation for another, a presentation generated before learning the verifier's fresh challenge cannot be reused successfully.

Compared to Schnorr-style proofs of knowledge, where freshness is typically obtained through a verifier-generated challenge, our construction achieves freshness with one fewer communication round. In a Schnorr-style protocol, the holder must first send a commitment, after which the verifier samples a fresh challenge and the holder responds. In our construction, no such initial commitment is required: the verifier samples a fresh unpredictable value aux and sends it to the holder, who then directly generates a presentation bound to aux.

2.2 Our Proof Techniques

In the unforgeability proof of our anonymous credential scheme, we want to show that an adversary cannot produce a valid fresh presentation for a set of revealed attributes Rev, on which the issuer has not issued a credential. Our proof proceeds directly by reduction to the q-DL assumption in the AGM.

Technical subtlety. The main difficulty is that a presentation reveals only Rev, while the polynomial underlying the credential, and hence the hidden attributes, is not given explicitly to the reduction. In anonymous credential schemes based on NIZKs, such hidden information can typically be recovered using the extractor of the underlying proof of knowledge. Our construction deliberately avoids NIZKs: validity is established only through algebraic relations between group elements. We must therefore show directly that the polynomials hidden in these relations can be extracted in the AGM.

Extraction lemmas. We prove two extraction lemmas, corresponding to the two pairing checks used by our construction. In these lemmas, the extractors recover the polynomial committed in the C.

The first lemma concerns blind issuance. Recall that an accepting issuance query $(C, \hat{C}, \mathsf{Rev})$ satisfies Equation (1), i.e., $\mathbf{e}(C, g_2) = e\left(h_1^{p_{\mathsf{Rev}}(\tau)}, \hat{C}\right)$. In the issuance-extraction lemma, i.e., Lemma 7.1, we show that, from any query satisfying this equation, a straight-line extractor can recover a polynomial p such that $C = h_1^{p(\tau)}$ and $p_{\mathsf{Rev}} \mid p$. This lemma allows us to replace the blind issuance oracle, which does not learn the polynomial being signed, by the simplified oracle in our unforgeability definition that receives p explicitly. Applying the extractor recursively to the adversary's issuance queries shows that the two formulations are equivalent.

The second is the presentation-extraction lemma, i.e., Lemma 7.2. For a fresh presentation, we consider Equation (3). We show that, whenever this equation holds and $(\mathsf{Rev}, \mathsf{aux})$ is fresh, we can extract a polynomial $p^\star$ such that $C = h_2^{p^\star(\tau)}$ and $p^\star_{\mathsf{Rev}} \mid p^\star$. Thus, although the adversary outputs only a presentation and never explicitly reveals the polynomial underlying its credential, the reduction can associate the forgery with an explicit polynomial $p^\star$.

The proofs of the extraction lemmas. Both lemmas follow the same algebraic proof strategy. For the issuance lemma, we substitute the adversary's AGM representations of C and $\hat{C}$ into Equation (1); for the presentation lemma, we substitute them into Equation (3). In either case, the pairing check induces a polynomial relation in the unknown setup values, secret-key shares, and oracle randomness.

We then distinguish two cases. If the induced relation is not the zero polynomial, we use it to break q-DL. We embed the q-DL challenge into the unknown exponents through independently randomized affine maps, as discussed in Lemma 3.1. Except with negligible probability, the resulting univariate polynomial remains nonzero and has the challenge value as a root, allowing the reduction to recover the challenge by polynomial factorization. If, on the other hand, the relation is identically zero, we show that algebraic independence forces coefficients of uniquely occurring monomials to vanish. The surviving coefficients then give exactly the polynomial encoded in C.

Adaptive corruptions. The extraction arguments require additional care because issuance and corruption queries may be adaptively interleaved. The crucial step is the zero-polynomial case of the extraction argument. After substituting the AGM representations of C and $\hat{C}$ into the pairing equation, we obtain a polynomial identity in the setup variables, the secret-key shares, and the randomness $\{r_i\}$ used in previous issuance queries. To extract the polynomial encoded in C, we must first rule out that C was constructed using previous issuance responses.

This is where the placement of C and $\hat{C}$ in different source groups becomes essential. A previous partial issuance response contains elements in $\mathbb{G}_1$ whose exponents involve terms of the form $\mathsf{sk}_i + \nu r_i p_i(\tau)$. If issuer i is corrupted only after this response was obtained, the adversary can remove the sk_i contribution and thereby reuse a $\mathbb{G}_1$ element whose exponent still depends on the previous issuance randomness r_i. Since $\hat{C}$ lies in $\mathbb{G}_2$, however, such an element cannot also occur in the AGM representation of $\hat{C}$. Consequently, the corresponding products of variables involving r_i occur uniquely in the polynomial identity induced by the pairing check. If that identity is the zero polynomial, the coefficients of these unique monomials must therefore be zero. This forces all contributions from previous issuance responses out of the representation of C, leaving only the CRS terms from which the extractor can compute the committed polynomial.

The remainder of the proof. With the extraction lemmas in place, every issuance query is associated with an explicit polynomial p_i, and the forged presentation with an explicit polynomial $p^\star$. We then apply essentially the same algebraic proof strategy as above, this time to the signature verification equation

$$e(\sigma_1^\star, g_2) = \mathsf{vk} \cdot e(\sigma_2^\star, h_2^{p^\star(\tau)}).$$

Substituting the AGM representations of the forged signature components yields a polynomial relation in the setup variables, the issuance randomness, and the secret-key shares.

As before, we distinguish whether this relation is the zero polynomial. If it is nonzero, our randomized embedding turns it into a nontrivial polynomial vanishing at the q-DL challenge, from which we recover

λ	Security parameter	m	Attribute	BG	Bilinear group
pp	Public parameters	M	Attribute Set	q_I	# issuance queries
l	Number of attributes	Rev	Set of revealed attributes	q_P	# presentation queries
sk	Signing key	Hid	Set of hidden attributes	aux	auxiliary data
vk	Verification key	σ	Signature/Credential		

Table 2: A summary of the notation used in this work.

the challenge. It therefore remains to rule out the zero-polynomial case. Cancelling the unique monomials first eliminates all terms that cannot occur in an identity; in particular, issuance queries on polynomials different from $p^\star$ disappear. The remaining terms contain the master secret sk together with secret-key shares sk_i corresponding only to issuers that were corrupted or issued on $p^\star$. By the winning condition, there are fewer than t such shares. The privacy of Shamir secret sharing then implies that these shares cannot algebraically determine, and hence cannot cancel, the sk. It follows that the relation cannot be identically zero.

3 Preliminaries

In this section, we describe the building blocks and preliminaries that our construction builds upon.

3.1 Notation

We denote vectors by bold-face letters, e.g. $\mathbf{x} = (x_1, \ldots, x_n)$, and sets by capital letters, e.g. M. We use $[n]$ to denote the list $[1, \ldots, n]$ and $l[i]$ to access the i-th index of list l. Further, we use $y \leftarrow F(x)$ to denote that y is assigned the output of algorithm F on input x. All algorithms referenced in our work are considered to be probabilistic and run in polynomial time (PPT). We write $y \leftarrow \langle A(x_1), B(x_2)\rangle$ to denote the execution of an interactive protocol between algorithm A on input x_1 and algorithm B on input x_2, where y refers to the protocol output. We use $y \xleftarrow{\$} Z$ to denote that y is sampled uniformly at random from set Z and use the same notation for outputs of randomized algorithms. We denote a t-out-of-n Shamir secret sharing [50] of a value x by $\mathsf{Shamir.SS}(x, n, t)$. We let $\lambda_{i,n}$ denote the ith Lagrange coefficient with respect to n. Further, we assume all adversaries to keep an internal state that is not given explicitly as input. We use the notation $\mathsf{Adv}^{\mathsf{Game}}_{\mathcal{A},\Pi}$ to denote the advantage of an adversary $\mathcal{A}$ in winning a security game Game instantiated with scheme Π. When considering the game for an arbitrary scheme, we may omit the scheme as a parameter. We summarize the meaning of all variables used in our work in Table 2.

Representation of attributes. We consider attribute sets $M \subseteq \mathbb{F}_p$ of size at most l. We associate to every attribute set M its canonical attribute polynomial

$$p_M(X) := \prod_{m \in M} (X - m) \in \mathbb{F}_p[X].$$

For a polynomial $p \in \mathbb{F}_p[X]$, we denote by $\textsc{Roots}(p) := \{x \in \mathbb{F}_p : p(x) = 0\}$ the set of its distinct roots in $\mathbb{F}_p$. In particular, $\textsc{Roots}(p_M) = M$. If p has no roots in $\mathbb{F}_p$, then $\textsc{Roots}(p) = \emptyset$.

3.2 Bilinear Maps

Let $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ be groups of prime order p and let g_1, g_2, and $g_T := \mathbf{e}(g_1, g_2)$ be generators of $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ respectively. A bilinear map $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable function that fulfills bilinearity and non-degeneracy. Throughout this paper, we use multiplicative notation for all groups. We define a bilinear group generator $\mathsf{BGGen}(1^\lambda)$ that outputs a group description $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$ for the security parameter λ.

3.3 Assumptions

Assumption 3.1 (q-DL). *For a bilinear group generator* BGGen*, we say the bilinear q discrete logarithm assumption [33] holds, if for any PPT adversary $\mathcal{A}$, and any $\lambda \in \mathbb{N}$, it holds that:*

$$\mathsf{Adv}^{\text{q-DL}}_{\mathsf{BGGen},\mathcal{A}}(\lambda) = \Pr\left[x' = x \;\middle|\; \begin{array}{c} \mathsf{BG} \leftarrow \mathsf{BGGen}(1^\lambda);\ x \xleftarrow{\$} \mathbb{F}_p \\ x' \xleftarrow{\$} \mathcal{A}\left(1^\lambda, \mathsf{BG}, \{g_1^{x^i}\}_{i\in[q]}, \{g_2^{x^i}\}_{i\in[q]}\right) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

3.4 Algebraic Group Model

The algebraic group model (AGM) defined by Fuchsbauer et al. [33] is a formal model in which algorithms are assumed to be algebraic in the sense that they can only perform algebraic operations on previously seen group elements. For every group element X output by an algorithm $\mathcal{A}$, it also provides the algebraic representation of X as a coefficient vector $\mathbf{z}$ such that $X = \prod_{i\in[n]} X[i]^{\mathbf{z}[i]}$, where $X[i]$ refers to the group elements that $\mathcal{A}$ has obtained before.

We will use the following lemma from [34] in our AGM security proofs.

Lemma 3.1 (q-DL Embedding [34]). *Let f be a non-zero multivariate polynomial in $\mathbb{F}_p[x_1, \ldots, x_m]$ of total degree d. Let $h(y) \in (\mathbb{F}_p[w_1, \ldots, w_m, v_1, \ldots, v_m])[y]$ be defined as*

$$h(y) := f(w_1 y + v_1, \ldots, w_m y + v_m),$$

Then the coefficient of maximal degree of h is a polynomial in $w_1, \ldots, w_m$ of degree d.

3.5 Threshold (Blind) Signature Schemes

We consider threshold (blind) signature schemes on message sets, parameterized by an upper-bound on the set size l, the number of issuers n and the threshold t. We first define traditional threshold signatures, and then give the adapted definition for blind signatures.

Definition 3.1 (Threshold Signature Scheme). *A threshold signature can be described as a tuple of the following algorithms:*

- $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$*: The setup algorithm takes the security parameter λ, the number of issuers n, and the number of messages l as input and outputs public parameters* pp*.*
- $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$*: The key-generation algorithm takes the public parameters* pp *and a threshold value t as input and outputs a set of signing and corresponding verification keys $\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}$, as well as the common verification key* vk*.*
- $\sigma_i \xleftarrow{\$} \mathsf{PSign}(\mathsf{pp}, \mathsf{sk}_i, M)$*: The partial signing algorithm takes a signing key sk_i and a message set M as input and outputs a partial signature σ_i on M.*
- $0/1 \xleftarrow{\$} \mathsf{PVerify}(\mathsf{pp}, \mathsf{vk}_i, \sigma_i, M)$*: The partial verification algorithm checks if a partial signature σ_i is valid for a message set M and a verification key vk_i.*
- $\sigma \xleftarrow{\$} \mathsf{Agg}(\mathsf{pp}, \{\sigma_i\}_{i\in S})$*: The aggregation algorithm takes a set of partial signatures as input and produces an aggregated signature σ.*
- $0/1 \leftarrow \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \sigma, M)$*: The verification algorithm outputs a bit indicating the validity of a signature σ for a message set M under the verification key* vk*.*

Threshold signature schemes must fulfill *correctness* (Definition 3.3) and *unforgeability* (Definition 3.4), intuitively meaning that honestly produced signatures must verify, and no efficient adversary should be able to produce a fresh signature without controlling at least t signers. We give the formal definitions of these properties below.

Originally introduced in [21], blind signatures allow a user to obtain a signature on messages hidden from the signer. We consider a variant of blind signatures in which the user can selectively reveal a subset of messages to the signer.

Definition 3.2 (Threshold Blind Signature Scheme). *A threshold blind signature scheme is a tuple of algorithms* (Setup, KeyGen, PSign, Obtain, PVerify, Agg, Verify)*, where* Setup*,* KeyGen*,* PVerify*,* Agg*, and* Verify *are defined as in Definition 3.1, and* PSign *and* Obtain *form an interactive blind partial signing protocol:*

- $\sigma_i \xleftarrow{\$} \langle \mathsf{PSign}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, \mathsf{Rev}) \rangle$*: The blind partial signing protocol between a user and signer i outputs a partial signature σ_i on message set M, while selectively revealing a subset* $\mathsf{Rev} \subseteq M$ *of messages to the signer.*

Threshold blind signatures must fulfill *correctness* (Definition 3.5), *one-more unforgeability (*OMUF*)* (Definition 3.6), and *blindness* (Definition 3.7). By correctness, honestly issued blind signatures must verify. One-more unforgeability ensures that an adversary cannot produce more signatures on unique messages than it has queried, and blindness ensures that an adversary cannot link blind signatures back to their signing sessions.

Definition 3.3 (Threshold Signature Correctness)**.** *We say that a threshold signature scheme is correct, if for any $\lambda, n, l \in \mathbb{N}$, any threshold $t \in [n]$, any $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$, any $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$, any $M \subseteq \mathbb{F}_p, |M| \leq l$, and any $S \subseteq [n]$ with $|S| \geq t$, it holds that:*

$$\Pr\left[\begin{array}{c|c} \forall i \in S : \mathsf{PVerify}(\mathsf{pp}, \mathsf{vk}_i, \sigma_i, M) = 1 & \forall i \in S : \sigma_i \xleftarrow{\$} \mathsf{PSign}(\mathsf{pp}, \mathsf{sk}_i, M) \\ \wedge\ \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \sigma, M) = 1 & \sigma \leftarrow \mathsf{Agg}(\mathsf{pp}, \{\sigma_i\}_{i\in S}) \end{array}\right] = 1.$$

Definition 3.4 (Threshold Existential Unforgeability)**.** *We say that a threshold signature scheme is existentially unforgeable under chosen-message attacks and adaptive corruptions* (tEUF-CMA)*, if for all $\lambda, n, l \in \mathbb{N}$, and all PPT adversaries $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathsf{tEUF\text{-}CMA}}_{\mathcal{A}}(\lambda, n, l) := \Pr\left[\mathsf{tEUF\text{-}CMA}_{\mathcal{A}}(\lambda, n, l) = 1\right] \leq \mathsf{negl}(\lambda),$$

where tEUF-CMA *is defined in Figure 1.*

$\mathsf{tEUF\text{-}CMA}_{\mathcal{A}}(\lambda, n, l)$:

1 : $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$
2 : $t \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{pp})$
3 : $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$
4 : $\mathcal{U}, \mathcal{C} \leftarrow \emptyset$
5 : $(M^*, \sigma^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\mathsf{PSign}}, \mathcal{O}^{\mathsf{Corr}}}(\{\mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk})$
6 : **return** $\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \sigma^*, M^*)$
$\wedge\ |\mathcal{C} \cup \mathcal{U}[M^*]| < t$

$\mathsf{OMUF}_{\mathcal{A}}(\lambda, n, l)$:

1 : $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$
2 : $t \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{pp})$
3 : $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$
4 : $\mathcal{C} \leftarrow \emptyset,\ q \leftarrow 0$
5 : $F \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\mathsf{PBSign}}, \mathcal{O}^{\mathsf{Corr}}}(\{\mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk})$
6 : **if** $|\mathcal{C}| \geq t$
return 0
7 : $k \leftarrow \lfloor q/(t - |\mathcal{C}|) \rfloor$ // #σs from q queries
8 : $(M_i^*, \sigma_i^*)_{i\in[k+1]} := F$
9 : **return** $\forall i, j \in [k+1], i \neq j : M_i^* \neq M_j^*$
$\wedge\ \forall i \in [k+1] \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \sigma_i^*, M_i^*)$

$\mathcal{O}^{\mathsf{PSign}}(M, i)$:

1 : $\sigma_i \xleftarrow{\$} \mathsf{PSign}(\mathsf{pp}, \mathsf{sk}_i, M)$
2 : $\mathcal{U}[M] \leftarrow \mathcal{U}[M] \cup \{i\}$
3 : **return** σ_i

$\mathcal{O}^{\mathsf{PBSign}}_{\mathcal{A}}(i, \mathsf{Rev})$:

1 : **if** $i \in \mathcal{C}$ **then return** $\perp$
2 : $\sigma_i \xleftarrow{\$} \langle \mathsf{PSign}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}), \mathcal{A}() \rangle$
$q \leftarrow q + 1$
3 : **return** σ_i

$\mathcal{O}^{\mathsf{Corr}}(i)$:

1 : $\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$
2 : **return** sk_i

Figure 1: tEUF-CMA game for threshold signatures and OMUF game for threshold blind signatures. Both games give $\mathcal{A}$ access to the corruption oracle $\mathcal{O}^{\mathsf{Corr}}$.

Definition 3.5 (Threshold Blind Signature Correctness)**.** *We say that a threshold blind signature scheme is correct, if for any $\lambda, n, l \in \mathbb{N}$, any threshold $t \in [n]$, any $M \subseteq \mathbb{F}_p, |M| \leq l$, any $\mathsf{Rev} \subseteq M$, and any $S \subseteq [n]$ with $|S| \geq t$,*

$$\Pr\left[\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \sigma, M) = 1 \;\middle|\; \begin{array}{c} \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l) \\ (\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t) \\ \forall i \in S : \sigma_i \xleftarrow{\$} \langle \mathsf{PSign}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}), \\ \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, \mathsf{Rev}) \rangle \\ \sigma \leftarrow \mathsf{Agg}(\mathsf{pp}, \{\sigma_i\}_{i\in S}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

For threshold one more unforgeability, the adversary $\mathcal{A}$ is tasked with outputting a number of signature forgeries, after getting access to a partial blind signing and corruption oracle. We let k be the number of valid signatures $\mathcal{A}$ can obtain from performing q partial signing queries. When corrupting $|\mathcal{C}|$ parties, where $|\mathcal{C}| < t$, $\mathcal{A}$ needs at least $t - |\mathcal{C}|$ partial signing queries to produce a full signature. We therefore find that $k \leftarrow \lfloor q/(t - |C|) \rfloor$ and require $\mathcal{A}$ to produce at least $k + 1$ forgeries. One can strengthen the definition by determining k via a more rigorous counting argument [40].

Definition 3.6 (Threshold One-More Unforgeability)**.** *We call a threshold blind signature scheme one-more unforgeable (*OMUF*), if for all PPT adversaries $\mathcal{A}$ and all $\lambda, n, l \in \mathbb{N}$,*

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathcal{A}}(\lambda, n, l) := \Pr\left[\mathsf{OMUF}_{\mathcal{A}}(\lambda, n, l) = 1\right] \leq \mathsf{negl}(\lambda),$$

where OMUF *is defined in Figure 1.*

For our definition of blindness, we consider an adversary that gets to choose two message sets. The challenger samples a secret bit b, which determines the order of issuance sessions for both message sets. After participating in both interactions and seeing the resulting credentials in a fixed order, the adversary must guess b'. This is consistent with blindness definitions in the literature, e.g., [40].

Definition 3.7 (Threshold Blindness)**.** *We call a threshold blind signature scheme blind, if for all adversaries $\mathcal{A}$ and all $\lambda, n, l \in \mathbb{N}$,*

$$\mathsf{Adv}^{\mathsf{tBLIND}}_{\mathcal{A}}(\lambda, n, l) := \Pr\left[\mathsf{tBLIND}_{\mathcal{A}}(\lambda, n, l) = 1\right] - \frac{1}{2} \leq \mathsf{negl}(\lambda),$$

where tBLIND *is defined in Figure 2.*

$\mathsf{tBLIND}_{\mathcal{A}}(\lambda, n, l)$:

1 : $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$
2 : $(t, \mathsf{vk}, \{\mathsf{vk}_i\}_{i\in[n]}, S, \{M_i\}_{i\in\{0,1\}}, \mathsf{Rev}) \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{pp})$
3 : $b \xleftarrow{\$} \{0, 1\}$
4 : **for** $x \in \{b, 1 - b\}$
5 : **for** $i \in S$ **do**
6 : $\sigma_{x,i} \leftarrow \langle \mathcal{A}(), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M_x, \mathsf{Rev})\rangle$
7 : $\sigma_x \leftarrow \mathsf{Agg}(\mathsf{pp}, \{\sigma_{x,i}\}_{i\in S})$
8 : $b' \xleftarrow{\$} \mathcal{A}(\sigma_0, \sigma_1)$
9 : **return** $(b' = b) \wedge (\mathsf{Rev} \subseteq M_0 \cap M_1) \wedge \sigma_0, \sigma_1 \neq \bot \wedge S \subseteq [n]$

Figure 2: Blindness (tBLIND) game for threshold blind signature schemes.

4 Anonymous Credentials

In this section, we give a formal definition of (threshold) anonymous credential schemes. We consider the classical model for anonymous credentials, with three types of parties: a number of n *issuers*, who issue (partial) credentials, credential *holders*, who obtain credentials, and present them to a *verifier*.

We define anonymous credentials on sets of attributes M. When issuing or presenting a credential, a holder can optionally reveal a subset of attributes, which we denote by Rev. Similarly, we denote the set of hidden attributes by $\mathsf{Hid} = M \setminus \mathsf{Rev}$. Our definition is parameterized by a space $\mathcal{X}_{\mathsf{aux}}$, from which auxiliary data can be sampled to be included in credential presentations. This is done to prevent replay attacks and to bind presentations to a session or context. Only auxiliary data can be drawn from $\mathcal{X}_{\mathsf{aux}}$, while credential attributes cannot be elements of $\mathcal{X}_{\mathsf{aux}}$.

4.1 Threshold Anonymous Credentials

As for threshold signatures (Definition 3.1), we let n be the number of issuers and denote the threshold by t.

Definition 4.1 (tAC Scheme)**.** *A threshold anonymous credential (*tAC*) scheme is a tuple of the following algorithms:*

- $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$*: The setup algorithm generates public parameters* pp *for a maximum number of* l *attributes.*
- $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$*: The key-generation algorithm takes the public parameters* pp *and a threshold* t *as input and outputs a set of signing and corresponding verification keys* $\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}$*, as well as the common verification key* vk*.*
- $\sigma_i \xleftarrow{\$} \langle \mathsf{PIssue}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, \mathsf{Rev})\rangle$*: The interactive partial (blind) issuance protocol between a holder and issuer* i *outputs a partial credential* σ_i *on an attribute set* M *to the holder, while selectively revealing a subset* $\mathsf{Rev} \subseteq M$ *of attributes to the issuer. The issuer takes its signing key* sk_i *and the revealed attributes* Rev *as input. The holder's* Obtain *algorithm takes the issuer's verification key* vk_i*, the attribute set* M *and the set* Rev *as input.*
- $\sigma \leftarrow \mathsf{Agg}(\{\sigma_i\}_{i\in S})$*: The aggregation algorithm takes a set of partial credentials* $\{\sigma_i\}_{i\in S}$ *as input and outputs an aggregated credential* σ*.*
- $0/1 \xleftarrow{\$} \langle \mathsf{Present}(\mathsf{pp}, \sigma, M, \mathsf{Rev}, \mathsf{aux}), \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{Rev}, \mathsf{aux})\rangle$*: The presentation is an interactive protocol between a holder running* $\mathsf{Present}$ *to present a credential* σ *while selectively revealing a subset* $\mathsf{Rev} \subseteq M$ *of attributes to a verifier running* Verify*. The verifier outputs* 1 *if the presentation is valid for a verification key* vk*, and auxiliary data* $\mathsf{aux} \in \mathcal{X}_{\mathsf{aux}}$*. Otherwise, it outputs* 0*.*
- $0/1 \leftarrow \mathsf{PVerify}(\mathsf{pp}, \mathsf{vk}_i, \sigma_i, p)$*: The partial verification algorithm takes the public parameters* pp*, an issuer verification key* vk_i*, a partial credential* σ_i*, and a polynomial* $p \in \mathbb{F}_p[X]$ *as input. It outputs* 1 *if* σ_i *is a valid partial credential on* p *under* vk_i*, and outputs* 0 *otherwise.*

4.2 tAC Security Properties

We call a tAC scheme *secure*, if it fulfills tAC-*Correctness* (Definition 4.2), tAC-*Unforgeability* (Definition 4.3), and tAC-*Unlinkability* (Definition 4.4). To define these properties, we adopt the game-based models of [42, 33, 23], extending them with a presentation oracle to explicitly model security against replay attacks.

Intuitively, correctness means that an honest holder can present a subset of the attributes of an honestly issued credential to a verifier, who will always accept. Unforgeability ensures that a malicious credential holder cannot generate a presentation for a set of attributes on which no credential was issued.

Unlinkability guarantees that a colluding issuer and verifier cannot link multiple presentations of the same credential to each other or to the issuance session.

For threshold schemes, an adversary obtains additional access to a corruption oracle allowing for adaptive corruptions. In the unlinkability game, the adversary is even assumed to compromise all parties.

Definition 4.2 (tAC-Correctness)**.** *A* tAC *scheme is correct, if for all* $\lambda, n, l, t \in [n]$*,* $M \subset \mathbb{F}_p, |M| \leq l$*,* $\mathsf{Rev}_I, \mathsf{Rev} \subseteq M$*,* $S \subseteq [n]$*,* $|S| \geq t$*,* $\mathsf{aux} \in \mathcal{X}_{\mathsf{aux}}$*, for*

$$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l), \quad (\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t),$$
$$\forall i \in S : \sigma_i \xleftarrow{\$} \langle \mathsf{PIssue}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}_I), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, \mathsf{Rev}_I)\rangle, \quad \sigma \leftarrow \mathsf{Agg}(\{\sigma_i\}_{i\in S})$$

it holds that

$$\Pr\left[\langle \mathsf{Present}(\mathsf{pp}, \sigma, M, \mathsf{Rev}, \mathsf{aux}), \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{Rev}, \mathsf{aux})\rangle = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

Our definition of unforgeability follows the established definitions for threshold anonymous credentials in the literature. An adversary $\mathcal{A}$, playing as the credential holder, gets access to a partial issuance oracle $\mathcal{O}^{\mathsf{PIssue}}$. After a number of queries, $\mathcal{A}$ must output a set of revealed attributes Rev^*, for which it can generate a valid presentation. The output is considered a forgery, if Rev^* is a subset of none of the attribute sets sent to $\mathcal{O}^{\mathsf{PIssue}}$. To model resilience against replay attacks, our definition additionally lets $\mathcal{A}$ query a (threshold) presentation oracle. The presentations are generated with respect to an attribute set M, revealed attributes Rev, and auxiliary data aux. When generating a forgery, $\mathcal{A}$ now also outputs auxiliary data aux^*, with respect to which the presentation must be valid. The challenger checks if the tuple $(\mathsf{Rev}^*, \mathsf{aux}^*)$ has been queried to $\mathcal{O}^{\mathsf{Present}}$ before. This opens a new attack vector for $\mathcal{A}$, as they can win the game by obtaining a presentation and presenting it for different auxiliary data. We give a formal definition of this property below.

Definition 4.3 (tAC-Unforgeability)**.** *We say a* tAC *scheme is unforgeable, if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function* negl*, and all* $\lambda, n, l \in \mathbb{N}$*, such that*

$$\mathsf{Adv}^{\mathsf{tAC\text{-}UF}}_{\mathcal{A}}(\lambda, n, l) = \Pr\left[\mathsf{tAC\text{-}UF}_{\mathcal{A}}(\lambda, n, l) = 1\right] \leq \mathsf{negl}(\lambda)$$

where Game tAC-UF *is defined in Figure 3.*

Game $\mathsf{tAC\text{-}UF}_{\mathcal{A}}(\lambda, n, l)$:

$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$
$t \xleftarrow{\$} \mathcal{A}(\mathsf{pp})$
$\left(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}\right) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$
$\mathcal{U}_{\mathsf{iss}}, \mathcal{U}_{\mathsf{pres}}, \mathcal{C} \leftarrow \emptyset$
$(\mathsf{Rev}^*, \mathsf{aux}^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\mathsf{PIssue}}, \mathcal{O}^{\mathsf{tPresent}}, \mathcal{O}^{\mathsf{Corr}}}(1^\lambda, \{\mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk})$
return $\langle \mathcal{A}(), \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{Rev}^*, \mathsf{aux}^*)\rangle$
$\quad \wedge (\mathsf{Rev}^*, \mathsf{aux}^*) \notin \mathcal{U}_{\mathsf{pres}} \wedge |\mathcal{C}| < t \wedge \mathsf{Rev}^* \cap \mathcal{X}_{\mathsf{aux}} = \emptyset \wedge \mathsf{aux}^* \in \mathcal{X}_{\mathsf{aux}}$
$\quad \wedge \forall M \in \mathcal{U}_{\mathsf{iss}} : |\mathcal{C} \cup \mathcal{U}_{\mathsf{iss}}[M]| < t \vee \mathsf{Rev}^* \not\subseteq M$

$\mathcal{O}^{\mathsf{PIssue}}_{\mathcal{A}}(p, i)$:

$M = \textsc{Roots}(p)$
if $i \in \mathcal{C}$:
$\quad$ **return** $\perp$
$\sigma \xleftarrow{\$} \langle \mathsf{PIssue}(\mathsf{pp}, \mathsf{sk}_i, M), \mathcal{A}()\rangle$
if $\mathsf{PVerify}(\mathsf{pp}, \mathsf{vk}_i, \sigma, p) = 0$
$\quad$ **return** $\perp$
$\mathcal{U}_{\mathsf{iss}}[M] \leftarrow \mathcal{U}_{\mathsf{iss}}[M] \cup \{i\}$
return σ

$\mathcal{O}^{\mathsf{tPresent}}_{\mathcal{A}}(M, \mathsf{Rev}, \mathsf{aux}, S)$:

if $M \cap \mathcal{X}_{\mathsf{aux}} \neq \emptyset \vee \mathsf{aux} \notin \mathcal{X}_{\mathsf{aux}} \vee |S| < t \vee S \cap \mathcal{C} \neq \emptyset$:
$\quad$ **return** $\perp$
for $i \in S$:
$\quad \sigma_i \xleftarrow{\$} \langle \mathsf{PIssue}(\mathsf{pp}, \mathsf{sk}_i, M), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, M)\rangle$
$\sigma \leftarrow \mathsf{Agg}(\{\sigma_i\}_{i\in S})$
$v \leftarrow \langle \mathsf{Present}(\mathsf{pp}, \sigma, M, \mathsf{Rev}, \mathsf{aux}), \mathcal{A}()\rangle$
$\mathcal{U}_{\mathsf{pres}} \leftarrow \mathcal{U}_{\mathsf{pres}} \cup \{(\mathsf{Rev}, \mathsf{aux})\}$
return v

$\mathcal{O}^{\mathsf{Corr}}(i)$:

$\mathcal{C} \leftarrow \mathcal{C} \cup \{i\}$
return sk_i

Figure 3: Unforgeability game tAC-UF for threshold anonymous credential (tAC) schemes.

Remark 4.1. *We simplify* tAC-UF *by disregarding blind issuance queries, as is common in the literature. This is justified by the extractability of partial issuance queries (Lemma 7.1). In our simplified definition, the adversary queries* $\mathcal{O}^{\mathsf{PIssue}}$ *directly on a polynomial* $p \in \mathbb{F}_p[X]$*, on which the oracle issues a partial credential.*

We do not require that p *has exactly* l *distinct roots in* $\mathbb{F}_p$*. In particular,* p *may have fewer than* l *roots, in which case only these roots correspond to the attributes. This does not cause a problem for the unforgeability definition: an issuance query on* p *is considered relevant to a presentation revealing* Rev *precisely if* $p_{\mathsf{Rev}} \mid p$*. Hence, the adversary can only use such an issuance query to present revealed attributes that actually occur as roots of* p*.*

For our unlinkability definition, we consider an adversary $\mathcal{A}$ controlling all issuers and verifiers, whose goal is to distinguish between issuance sessions and presentations of two distinct attribute sets. To this end, $\mathcal{A}$ chooses the secret and verification keys, outputs two attribute sets and then participates in the issuance and presentation protocol for one of them. $\mathcal{A}$ also has the ability to choose the set of attributes revealed during issuance Rev_I and presentation Rev, with the constraint that both are subsets of the two attribute sets. We give a formal definition below.

Definition 4.4 (tAC-Unlinkability)**.** *We say a* tAC *scheme is unlinkable, if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function* negl*, such that for all* $\lambda, n, l \in \mathbb{N}$*,*

$$\mathsf{Adv}^{\mathsf{tAC\text{-}UL}}_{\mathcal{A}}(\lambda, n, l) = Pr\left[\mathsf{tAC\text{-}UL}_{\mathcal{A}}(\lambda, n, l) = 1\right] - \frac{1}{2} \leq \mathsf{negl}(\lambda)$$

where Game tAC-UL *is defined in Figure 4.*

Game $\mathsf{tAC\text{-}UL}_{\mathcal{A}}(\lambda, n, l)$:

1 : $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$
2 : $(t, \mathsf{vk}, \{\mathsf{vk}_i\}_{i\in[n]}, \{M_i\}_{i\in\{0,1\}}, \mathsf{Rev}, \mathsf{Rev}_I, \mathsf{aux}) \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{pp})$
3 : $b \xleftarrow{\$} \{0,1\}$
4 : **for** $i \in [n]$:
5 : $\quad (\sigma_{b,i}) \leftarrow \langle \mathcal{A}(), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M_b, \mathsf{Rev}_I)\rangle$
6 : $\sigma_b \leftarrow \mathsf{Agg}(\{\sigma_{b,i}\}_{i\in[n]})$
7 : $v^* \xleftarrow{\$} \langle \mathsf{Present}(\mathsf{pp}, \sigma_b, M_b, \mathsf{Rev}, \mathsf{aux}), \mathcal{A}()\rangle$
8 : $b' \xleftarrow{\$} \mathcal{A}(v^*)$
9 : **return** $(b' = b) \wedge (\mathsf{Rev}, \mathsf{Rev}_I \subseteq M_0 \wedge \mathsf{Rev}, \mathsf{Rev}_I \subseteq M_1) \wedge \sigma_b \neq \bot$

Figure 4: Unlinkability game tAC-UL for threshold anonymous credential (tAC) schemes.

5 BB-KZG: Our modified threshold signature scheme

In this section, we introduce the modified threshold BB signature scheme that our credential construction is based on. As the construction encodes attribute sets similarly to KZG commitments, we name this construction BB-KZG. We first provide a description of the construction itself, then reduce its security to the q-DL assumption in the algebraic group model.

5.1 The BB-KZG Construction

As discussed in Section 2, we modify the BB signature scheme using KZG commitments. We present the complete description of this construction in Figure 5.

$\mathsf{Setup}_{\mathsf{BGGen}}(\lambda, n, l)$:

1 : $\tau, \nu \xleftarrow{\$} \mathbb{F}_p$
2 : $\mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(\lambda)$
3 : $h_1 \leftarrow g_1^\nu;\ h_2 \leftarrow g_2^\nu$
4 : **return** $\left(n, \mathsf{BG}, \left\{g_1^{\tau^i}, h_1^{\tau^i}, h_2^{\tau^i}\right\}_{i\in[0,l]}\right)$

$\mathsf{KeyGen}(\mathsf{pp}, t)$:

1 : $\mathsf{sk} \xleftarrow{\$} \mathbb{F}_p$
2 : $\{\mathsf{sk}_i\}_{i\in[n]} \xleftarrow{\$} \text{Shamir.SS}(\mathsf{sk}, n, t)$
3 : **return** $\left(\left\{g_1^{\mathsf{sk}_i}, g_T^{\mathsf{sk}_i}\right\}_{i\in[n]}, g_T^{\mathsf{sk}}\right)$

$\mathsf{PSign}(\mathsf{pp}, \mathsf{sk}_i, M)$:

1 : $r \xleftarrow{\$} \mathbb{F}_p$
2 : $\sigma_1 \leftarrow g_1^{\mathsf{sk}_i} \cdot h_1^{p_M(\tau)\cdot r}$
3 : $\sigma_2 \leftarrow g_1^r$
4 : **return** (σ_1, σ_2)

$\mathsf{Agg}(\mathsf{pp}, \{\sigma_{1,i}, \sigma_{2,i}\}_{i\in S})$:

1 : $\sigma_1 \leftarrow \prod_{i\in S} \sigma_{1,i}^{\lambda_{i,S}}$
2 : $\sigma_2 \leftarrow \prod_{i\in S} \sigma_{2,i}^{\lambda_{i,S}}$
3 : **return** (σ_1, σ_2)

$\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \sigma, M)$:

1 : $(\sigma_1, \sigma_2) := \sigma$
2 : **return** $\mathbf{e}(\sigma_1, g_2) \stackrel{?}{=} \mathsf{vk} \cdot \mathbf{e}(\sigma_2, h_2^{p_M(\tau)})$

Figure 5: Our BB-KZG construction, where $p_M(\tau) := \prod_{\mathrm{m}\in M}(\tau - \mathrm{m})$. We let $\mathsf{PVerify}(\mathsf{pp}, \mathsf{vk}_i, \sigma_i, p_M) = \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}_i, \sigma_i, M)$.

5.2 Security of BB-KZG

Correctness Note that the elements $h_1^{p_M(\tau)}$ and $h_2^{p_M(\tau)}$ can be computed from the setup using the elements $\{h_1^{\tau^i}, h_2^{\tau^i}\}_{i\in[0,l]}$. Correctness follows by inspection.

Theorem 5.1 (Unforgeability of BB-KZG)**.** *If the bilinear* q-DL *assumption holds,* BB-KZG *is* tEUF-CMA *secure as per Definition 3.4 in the algebraic group model.*

The proof formalizes the arguments discussed in Section 2. We defer it to Appendix A.

6 A Blind Variant of BB-KZG

We next give the full details of our blinded variant of BB-KZG, which we call Blind-BB-KZG. Our construction supports partial blind issuance, meaning that a user holding a set of attributes M can choose to reveal a subset Rev to the signer, while keeping the remaining attributes Hid hidden. We refer to Section 2 for a high-level description of the scheme.

6.1 The blinded Construction: Blind-BB-KZG

We present the blinded construction in the following.

The setup algorithm now additionally outputs $\{g_2^{\tau^i}\}_{i\in[0,l]}$, while the key generation, aggregation, and verification algorithms are as in BB-KZG.

Sign - $\langle \mathsf{PBSign}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, \mathsf{Rev})\rangle$**:** The interactive partial blind signing protocol selectively reveals a subset of attributes $\mathsf{Rev} \subseteq M$ to signer i.

1. Let

$$p_M(\tau) \coloneqq \prod_{\mathrm{m}\in M} (\tau - \mathrm{m}), \quad \text{and} \quad p_{\mathsf{Hid}}(\tau) \coloneqq \prod_{\mathrm{m}\in \mathsf{Hid}} (\tau - \mathrm{m}),$$

   where $\mathsf{Hid} = M \setminus \mathsf{Rev}$. If $g_1^{\mathrm{m}} = g_1^{\tau}$ for some $\mathrm{m} \in M$, the user aborts and outputs $\perp$. The user samples $a \xleftarrow{\$} \mathbb{F}_p^*$ and computes

$$C \leftarrow h_1^{a\cdot p_M(\tau)}, \quad \text{and} \quad \hat{C} \leftarrow g_2^{a\cdot p_{\mathsf{Hid}}(\tau)},$$

   which are computable using the $\{h_1^{\tau^i}\}_{i\in[l]}$ terms included in pp. It sends $(C, \hat{C})$ to the signer.

2. Let

$$p_{\mathsf{Rev}}(\tau) \coloneqq \prod_{\mathrm{m}\in \mathsf{Rev}} (\tau - \mathrm{m}).$$

   The signer first checks that $C, \hat{C} \neq 1$ and that

$$\mathbf{e}(C, g_2) \stackrel{?}{=} \mathbf{e}\left(h_1^{p_{\mathsf{Rev}}(\tau)}, \hat{C}\right). \tag{4}$$

   It proceeds as in BB-KZG by sampling $r \xleftarrow{\$} \mathbb{F}_p$ and computing

$$\sigma_1' \leftarrow g_1^{\mathsf{sk}_i} C^r, \qquad \sigma_2' \leftarrow g_1^r,$$

   and sending $\sigma \coloneqq (\sigma_1', \sigma_2')$ to the user. Finally, the user samples new randomness $r' \xleftarrow{\$} \mathbb{F}_p$ and adapts the signature with the blinding factor as $\sigma_1 \leftarrow \sigma_1' \cdot C^{r'}$, and $\sigma_2 \leftarrow (\sigma_2' \cdot g_1^{r'})^a$ and stores $\sigma \coloneqq (\sigma_1, \sigma_2)$.

Remark 6.1. *The above construction can be proven secure for any $n \in \mathbb{N}$ and $t \in [n]$. For $n = t = 1$, a slightly more efficient construction would send both C and $\hat{C}$ in $\mathbb{G}_1$ for the blind issuance query. This variant remains secure while requiring a smaller CRS, however the extractability of issuance queries (Lemma 7.1) fails for thresholds $t > 1$, due to an attack in which an adaptive adversary sends a blind issuance query to an issuer it later corrupts.*

6.2 Security of Blind-BB-KZG

Correctness. It follows by inspection that

$$\mathbf{e}(\sigma_1, g_2) = \mathsf{vk}_i \cdot \mathbf{e}(h_1^{a p_M(\tau)(r+r')}, g_2) = \mathsf{vk}_i \cdot \mathbf{e}(g_1^{a(r+r')}, h_2^{p_M(\tau)}) = \mathsf{vk}_i \cdot \mathbf{e}(\sigma_2, h_2^{p_M(\tau)}).$$

Correctness fails only if $\tau \in M$, which happens with probability at most l/p.

Theorem 6.1 (Blindness of Blind-BB-KZG)**.** Blind-BB-KZG *fulfills blindness as per Definition 3.7.*

Intuitively, the scheme fulfills blindness, since C and $\hat{C}$ both are computed using uniformly sampled randomness. We show that the query does not reveal information about the hidden attributes it encodes, since a given query will be generated from two different hidden attributes with equal probability. Moreover, rerandomizing the signature components with fresh randomness r' prevents the issuer from linking it back to the signature it computed during issuance. We present the full proof in Appendix A.

Theorem 6.2 (One-more unforgeability of Blind-BB-KZG)**.** *If the bilinear* q-DL *assumption holds,* Blind-BB-KZG *is one-more unforgeable as per Definition 3.6 in the algebraic group model.*

This proof formalizes the arguments discussed in Section 2 and is presented in Appendix A. We note that the proof follows the same structure as the unforgeability proof of our credential construction in Theorem 7.4.

7 Constant-Size (Threshold) Anonymous Credentials

In this section, we present our threshold anonymous credential construction BBCred. The issuance protocol is the same as the blind signing protocol in Blind-BB-KZG. We additionally introduce a presentation protocol that randomizes the signature and proves knowledge of a subset of attributes. To prevent replay attacks, where an adversary could rerandomize a credential presentation and present it as their own, the presentation is bound to an auxiliary input aux. aux can be freshly generated for each presentation; for instance, it may consist of the verifier's identity and a nonce chosen by the verifier. We refer to Section 2 for an overview of the construction.

The BBCred Construction. We now present the full construction.

Setup - $\mathsf{Setup}_{\mathsf{BGGen}}(\lambda, n, l)$**:** Samples $\tau, \nu \xleftarrow{\$} \mathbb{F}_p$, computes $\mathsf{BG} \leftarrow \mathsf{BGGen}(\lambda), h_1 = g_1^{\nu}, h_2 = g_2^{\nu}$, and outputs the public parameters

$$\mathsf{pp} \leftarrow \left(n, \mathsf{BG}, h_1, h_2, \{g_1^{\tau^i}, g_2^{\tau^i}\}_{i\in[l+1]}, \{h_1^{\tau^i}, h_2^{\tau^i}\}_{i\in[l]}\right).$$

Note that this is the setup of Blind-BB-KZG except for the additional $g_1^{\tau^{l+1}}, g_2^{\tau^{l+1}}$ elements.

Key Generation - $\mathsf{KeyGen}(\mathsf{pp}, t)$**:** Same as in Blind-BB-KZG: samples $\mathsf{sk} \xleftarrow{\$} \mathbb{F}_p$, computes $\{\mathsf{sk}_i\}_{i\in[n]} \xleftarrow{\$}$ Shamir.SS(sk, n, t), and outputs $\left(\left\{g_1^{\mathsf{sk}_i}, g_T^{\mathsf{sk}_i}\right\}_{i\in[n]}, g_T^{\mathsf{sk}}\right)$.

Partial Verification - $\mathsf{PVerify}(\mathsf{pp}, \mathsf{vk}_i, (\sigma_1, \sigma_2), p)$: Checks if $\mathbf{e}(\sigma_1, g_2) \stackrel{?}{=} \mathsf{vk}_i \mathbf{e}(\sigma_2, h_2^{p(\tau)})$ and returns the result.

Issuance - $\langle \mathsf{PBIssue}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, \mathsf{Rev})\rangle$**:** Same as in Blind-BB-KZG: the holder and issuer i interactively compute[2]:

$$(\sigma_1, \sigma_2) \leftarrow \text{Blind-BB-KZG}.\langle \mathsf{PBSign}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, \mathsf{Rev})\rangle.$$

Aggregation - $\mathsf{Agg}(\mathsf{pp}, \{\sigma_{1,i}, \sigma_{2,i}\}_{i\in S})$: The holder computes

$$\sigma_1 \leftarrow \prod_{i\in S} \sigma_{1,i}^{\lambda_{i,S}}, \quad \sigma_2 \leftarrow \prod_{i\in S} \sigma_{2,i}^{\lambda_{i,S}},$$

and returns (σ_1, σ_2).

Presentation - $\langle \mathsf{Present}(\mathsf{pp}, \sigma, M, \mathsf{Rev}, \mathsf{aux}), \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{Rev}, \mathsf{aux})\rangle$: The holder samples $a_0 \xleftarrow{\$} \mathbb{F}_p$, $a_1 \xleftarrow{\$} \mathbb{F}_p^*$ and computes:

$$\sigma_1' \leftarrow \sigma_1 \cdot h_1^{a_0 \cdot p_M(\tau)}, \qquad \sigma_2' \leftarrow (\sigma_2 \cdot g_1^{a_0})^{\frac{1}{a_1}}.$$
$$C \leftarrow h_2^{a_1 \cdot p_M(\tau)}, \qquad \hat{C} \leftarrow g_1^{a_1 \cdot p_{\mathsf{Hid}}(\tau)\cdot(\tau - \mathsf{aux})},$$

[2]Note that in Blind-BB-KZG, the holder rerandomizes the signature after unblinding. This additional step is not needed here, since rerandomization is performed as part of the presentation protocol.

where p_M and p_{Hid} are defined as in the issuance protocol and contain roots for all elements in M or Hid. It then sends the presentation $(\sigma_1', \sigma_2', C, \hat{C})$ to the verifier. The verifier checks that

$$\mathbf{e}\left(g_1^{(\tau-\mathsf{aux})}, C\right) \stackrel{?}{=} \mathbf{e}\left(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}\right), \tag{5}$$

and

$$\mathbf{e}\left(\sigma_1', g_2\right) \stackrel{?}{=} \mathsf{vk} \cdot \mathbf{e}\left(\sigma_2', C\right). \tag{6}$$

If both checks hold, the verifier accepts. Otherwise it rejects.

Remark 7.1. *Note that multiplying $\hat{C}$ with $\tau - \mathsf{aux}$, conceptually treats aux as an additional attribute in $\hat{C}$. Hence, the presentation proof is bound to aux, since the verifier can only factor out roots of polynomial $p_{\mathsf{Hid}}(\tau) \cdot (\tau - \mathsf{aux})$.*

Remark 7.2. *For clearer presentation, our verification check is presented as two separate pairing checks. In practice, the verifier can save a pairing check by sampling a challenge $\gamma \xleftarrow{\$} \mathbb{F}_p$ and verifying*

$$\mathbf{e}(\sigma_1', g_2) \cdot \mathbf{e}\left(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}\right)^{\gamma} \stackrel{?}{=} \mathsf{vk} \cdot \mathbf{e}\left(\sigma_2' \cdot g_1^{\gamma(\tau-\mathsf{aux})}, C\right).$$

This is a standard technique for batching pairing checks [15], where γ enforces a random linear combination of the two checks such that the combined check holds if and only if both individual checks hold with overwhelming probability.

7.1 Security of BBCred

In this section, we show that our BBCred construction achieves the properties of a tAC scheme. We start by showing correctness.

Correctness. Correctness follows directly by verifying the presentation equations against the aggregated, randomized signature.

Theorem 7.1 (Correctness of BBCred)**.** *Our* BBCred *construction fulfills correctness as per Definition 4.2.*

Proof. We show that for all $\lambda, n, l, t \in \mathbb{N}$, $M \subset \mathbb{F}_p, |M| \leq l$, $\mathsf{Rev}, \mathsf{Rev}_I \subseteq M$, $S \subseteq [n]$, $\mathsf{aux} \in \mathcal{X}_{\mathsf{aux}}$, for

$$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l), \quad (\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$$
$$\sigma \xleftarrow{\$} \langle \mathsf{PIssue}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Rev}_I), \mathsf{Obtain}(\mathsf{pp}, \mathsf{vk}_i, M, \mathsf{Rev}_I)\rangle,$$

it holds that

$$\Pr\left[\langle \mathsf{Present}(\mathsf{pp}, \sigma, M, \mathsf{Rev}, \mathsf{aux}), \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \mathsf{Rev}, \mathsf{aux})\rangle = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

if $\tau \in M$, which happens with probability at most l/p, the holder aborts. For $i \in S$, let $\sigma_i = (\sigma_{1,i}, \sigma_{2,i})$ be a partial credential on attributes M as output by the partial $\mathsf{PBIssue}$ protocol:

$$\sigma_{1,i} = g_1^{\mathsf{sk}_i} \cdot h_1^{p_M(\tau)\cdot ar_i}, \quad \sigma_{2,i} \leftarrow g_1^{ar_i}$$

Aggregating the partial credentials, we have that

$$\sigma_1 = \prod_{i\in S} \sigma_{1,i}^{\lambda_{i,S}} = g_1^{\sum_{i\in S}\lambda_{i,S}\mathsf{sk}_i} \cdot h_1^{p_M(\tau)ar} = g_1^{\mathsf{sk}} \cdot h_1^{p_M(\tau)ar}, \qquad \sigma_2 = \prod_{i\in S} \sigma_{2,i}^{\lambda_{i,S}} = g_1^{ar},$$

where $r = \sum_{i\in S}\lambda_{i,S} r_i$. Let $r' = ar$. After randomization in the presentation protocol, we have that

$$\sigma_1' = \sigma_1 \cdot h_1^{a_0 \cdot p_M(\tau)} = g_1^{\mathsf{sk}} \cdot h_1^{p_M(\tau)\cdot(a_0+r')}, \qquad \sigma_2' = (\sigma_2 \cdot g_1^{a_0})^{a_1^{-1}} = g_1^{\frac{a_0+r'}{a_1}}.$$

Let $(\sigma_1', \sigma_2', C, \hat{C})$ be the presentation proof sent to the verifier. Verifying Equations (5) and (6), we find that

$$\mathbf{e}\left(g_1^{(\tau-\mathsf{aux})}, C\right) = \mathbf{e}\left(g_1^{a_1\cdot p_{\mathsf{Hid}}(\tau)\cdot(\tau-\mathsf{aux})}, h_2^{p_{\mathsf{Rev}}(\tau)}\right) = \mathbf{e}\left(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}\right)$$

and

$$\mathbf{e}(\sigma_1', g_2) = \mathsf{vk} \cdot \mathbf{e}\left(g_1^{(a_0+r')\cdot a_1^{-1}}, h_2^{a_1\cdot p_M(\tau)}\right) = \mathsf{vk} \cdot \mathbf{e}\left(\sigma_2', C\right).$$

□

Unlinkability. Similarly as for the blind signature scheme (Theorem 6.1), BBCred's issuance queries do not reveal information about the hidden attributes they encode. Further, we show that a given presentation can be generated from two different credentials containing the same set of revealed attributes with equal probability, by showing that there exists a bijective function that maps presentations of one credential to presentations of the other.

Theorem 7.2 (Unlinkability of BBCred)**.** *Our construction* BBCred *fulfills unlinkability as per Definition 4.4.*

We give the full proofs in Appendix B.

Unforgeability. We prove the tAC-UF security of BBCred in the algebraic group model. In the following, we will present all theorems and lemmas relevant to the proof. For a high-level overview of how our proofs proceed, we refer to Section 2. For the formal proofs, we refer to Appendix B.

First, we present a theorem justifying our simplified tAC-UF definition (Figure 3). We show that a partial blind issuance oracle $\mathcal{O}^{\mathsf{PBIssue}}$ can be simulated using access to an $\mathcal{O}^{\mathsf{PIssue}}$ oracle, which handles the game-specific bookkeeping. Let $\mathcal{O}^{\mathsf{PBIssue}}$ be defined as $\mathcal{O}^{\mathsf{PIssue}}$ (Figure 3), with the distinction that $\mathcal{O}^{\mathsf{PBIssue}}$ does not take the polynomial as input, extract the roots, call $\mathsf{PVerify}$ or update $\mathcal{U}_{\mathsf{iss}}$.

Theorem 7.3 (Blind Issuance Oracle Simulation)**.** *Let $\mathcal{A}$ be an algebraic PPT adversary in Game* tAC-UF *(Figure 3) with access to $\mathcal{O}^{\mathsf{PBIssue}}$ in place of $\mathcal{O}^{\mathsf{PIssue}}$. Let $(C_k, \hat{C}_k, \mathsf{Rev}_k)$ be the message of $\mathcal{A}$ in its k-th non-aborting query to $\mathcal{O}^{\mathsf{PBIssue}}(i_k)$. Under the bilinear* q-DL *assumption, there exists a PPT adversary $\mathcal{B}$ for* tAC-UF *that runs $\mathcal{A}$ such that, except with negligible probability,*

1. *$\mathcal{B}$ perfectly simulates the view of $\mathcal{A}$, and*
2. *$\mathcal{B}$ queries $\mathcal{O}^{\mathsf{PIssue}}$ exactly on the pairs (p_k, i_k), where $C_k = h_1^{p_k(\tau)}$ and $p_{\mathsf{Rev}_k} \mid p_k$.*

We then present a lemma establishing that partial blind issuance queries are extractable in the AGM. The lemma is used to prove Theorem 7.3 via induction.

Lemma 7.1 (Extractability of partial blind issuance queries)**.** *Let* BGGen *be a bilinear group generator. Let $\mathcal{A}$ be an algebraic PPT adversary that for $\lambda, n, l \in \mathbb{N}$ receives $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$, chooses $t \leq n$, and obtains vk as input, where $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$, and has access to oracles $\mathcal{O}^{\mathsf{PIssue}}, \mathcal{O}^{\mathsf{tPresent}}$ and $\mathcal{O}^{\mathsf{Corr}}$ for* BBCred *as per Definition 4.3. Assuming the bilinear* q-DL *assumption holds, if $\mathcal{A}$ outputs $(C, \hat{C}, \mathsf{Rev})$ such that $\mathbf{e}(C, g_2) = \mathbf{e}(h_1^{p_{\mathsf{Rev}}(\tau)}, \hat{C})$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ with the same input and access to $\mathcal{O}^{\mathsf{PIssue}}$, $\mathcal{O}^{\mathsf{tPresent}}$, $\mathcal{O}^{\mathsf{Corr}}$, and $\mathcal{A}$, that outputs a polynomial p, such that $C = h_1^{p(\tau)}$ and $p_{\mathsf{Rev}} | p$ with probability $1 - \mathsf{negl}(\lambda)$.*

Similarly as for the partial blind issuance queries, we present a lemma that establishes the extractability of credential presentations in the AGM, meaning that malicious holders cannot produce a valid presentation without knowing a corresponding credential.

Lemma 7.2 (Extractability of presentations)**.** *Let* BGGen *be a bilinear group generator. Let $\mathcal{A}$ be an algebraic PPT adversary that for $\lambda, n, l \in \mathbb{N}$ receives $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$ and vk as input, where $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$, and has access to oracles $\mathcal{O}^{\mathsf{PIssue}}$, $\mathcal{O}^{\mathsf{tPresent}}$, and $\mathcal{O}^{\mathsf{Corr}}$ for* BBCred *as per Definition 4.3. Assuming the bilinear* q-DL *assumption holds, if $\mathcal{A}$ outputs $(C, \hat{C}, \mathsf{aux}, \mathsf{Rev})$, such that $\mathbf{e}\left(g_1^{\tau - \mathsf{aux}}, C\right) = \mathbf{e}\left(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}\right)$ and $(\mathsf{Rev}, \mathsf{aux})$ has not been queried to $\mathcal{O}^{\mathsf{tPresent}}$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ with the same input and access to $\mathcal{O}^{\mathsf{PIssue}}$, $\mathcal{O}^{\mathsf{tPresent}}$ and $\mathcal{A}$, that outputs a polynomial p, such that $C = h_2^{p(\tau)}$ and $p_{\mathsf{Rev}} | p$ with probability $1 - \mathsf{negl}(\lambda)$.*

Finally, we state the unforgeability of BBCred with respect to our tAC-UF definition. The theorem is proven using the extractability of presentations (Lemma 7.2), and follows the strategy outlined in Section 2.

Theorem 7.4 (Unforgeability of BBCred)**.** *If the bilinear* q-DL *assumption holds, our credential construction* BBCred *fulfills unforgeability as per Definition 4.3 in the algebraic group model.*

We refer to Appendix B for the full proofs.

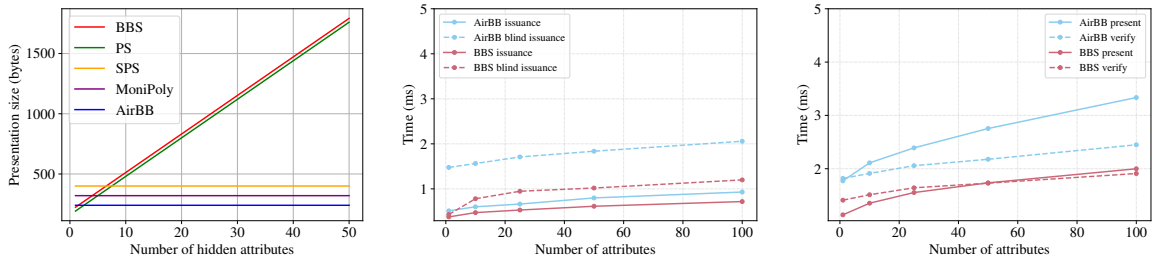

Figure 6: Comparison of BBCred against related work in terms of presentation size, issuance runtime (blind and non-blind), and presentation generation and verification time. For the issuance and presentation, half the total number of attributes are revealed. By BBS we refer to [53], by PS to [46], by SPS to [32], by MoniPoly to [52], and by AirBB to our work.

8 Evaluation

To demonstrate the practicality of our construction, we implement and benchmark BBCred, including its threshold variant. We implement the scheme in Rust on the BLS12-381 curve using the `arkworks` library[3]. Our source code including benchmark scripts is publicly available at `https://github.com/AppliedCryptoGroup/airbb-creds`. All benchmarks were performed on a MacBook Pro with an Apple M3 Max chip and 36 GB of RAM running macOS 26.4.1. Each measurement is the median of 100 independent runs. We compare against BBS [53] as our primary baseline. Recall that BBCred presentations have a constant size of 240 bytes, while BBS presentations grow linearly with the number of attributes.

Communication costs. In the left plot of Figure 6, we report the estimates of the presentation sizes of BBCred alongside related work for a varying number of hidden attributes. The cost estimates mirror the numbers given in Table 1 and were obtained from concrete figures in the paper (BBS and PS), and estimates based on the paper and known sizes of zero-knowledge proofs (SPS-EQ, MoniPoly). We assume a compressed representation of group elements, that is 48 bytes for $\mathbb{G}_1$ and 96 bytes for $\mathbb{G}_2$ elements. A BBCred presentation is independent of the number of hidden attributes, and consists of only three $\mathbb{G}_1$ elements and one $\mathbb{G}_2$ element, which results in a size of only 240 bytes overall. In contrast, for BBS, the presentation size grows linearly in the number of hidden attributes given by two $\mathbb{G}_1$ and $3 + |\mathsf{Hid}|$ field elements.

Computational costs. The middle plot of Figure 6 shows the computation time for the non-blind (i.e., with $\mathsf{Hid} = \emptyset$) and blind issuance of the single-issuer BBCred and BBS schemes with respect to the total number of attributes, while always revealing half. We further report the computation time for generating and verifying a presentation in the right plot of Figure 6, again for varying numbers of total attributes and always revealing half. The results confirm that BBCred is on par with BBS in terms of efficiency, achieving non-blind issuance times of under 1 ms and blind issuance times under 2.1 ms, as well as presentation generation and verification under 3.4 ms, all for credentials with up to 100 attributes. Note that for the non-blind issuance of BBCred, we rely on the standard signing algorithm as given in Section 5.1, which does not require a pairing check, thus explaining the overhead of ∼1 ms for blind issuance.

Threshold aggregation costs. Since partial signing, presentation, and verification match the non-threshold costs above, aggregating t partial signatures is the only added cost of thresholdization, and this cost depends only on how many shares are aggregated. Aggregation takes under 1.7 ms even for $t = 100$, confirming negligible overhead.

[3] https://github.com/arkworks-rs/algebra

Acknowledgements

The authors have used LLM-based tools to improve conciseness and phrasing. This work was partially supported by the European Research Council (ERC) under the European Union's Horizon Europe research and innovation program (grant CRYPTOLAYER-101044770), and by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts, through their joint sponsorship of the National Research Center for Applied Cybersecurity ATHENE.

References

[1] ISO/IEC 20008-2:2013 — information technology — security techniques — anonymous digital signatures — part 2: Mechanisms using a group public key. Standard ISO/IEC 20008-2:2013, International Organization for Standardization, Geneva, Switzerland (2013), https://www.iso.org/standard/56916.html 1

[2] Abe, M., Nanri, M., Perez-Kempner, O., Tibouchi, M.: Interactive threshold mercurial signatures and applications. In: ASIACRYPT (3). Lecture Notes in Computer Science, vol. 15486, pp. 69–103. Springer (2024) 4

[3] Ames, S., Hazay, C., Ishai, Y., Venkitasubramaniam, M.: Ligero: lightweight sublinear arguments without a trusted setup. Des. Codes Cryptogr. **91**(11), 3379–3424 (2023) 3

[4] Au, M.H., Susilo, W., Mu, Y.: Constant-size dynamic k-taa. In: Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings (2006). https://doi.org/10.1007/11832072_8 1, 2

[5] Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: Non-interactive anonymous credentials. IACR Cryptol. ePrint Arch. (2007) 3

[6] Bernstein, G., Sporny, M.: Data integrity BBS cryptosuites v1.0. W3C candidate recommendation draft, World Wide Web Consortium (2026), https://www.w3.org/TR/vc-di-bbs/ 1

[7] Boneh, D., Boyen, X.: Efficient selective-id secure identity-based encryption without random oracles. In: EUROCRYPT. Lecture Notes in Computer Science, Springer (2004) 2

[8] Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: CRYPTO. Lecture Notes in Computer Science, Springer (2004) 1, 2

[9] Bormann, C., Lehmann, A.: Sok: Anonymous credentials for digital identity wallets. IACR Cryptol. ePrint Arch. (2026) 1

[10] Brickell, E., Li, J.: Enhanced privacy ID: A direct anonymous attestation scheme with enhanced revocation capabilities. IEEE Trans. Dependable Secur. Comput. (2012) 3

[11] Camenisch, J., Chen, L., Drijvers, M., Lehmann, A., Novick, D., Urian, R.: One TPM to bind them all: Fixing TPM 2.0 for provably secure anonymous attestation. In: IEEE Symposium on Security and Privacy. IEEE Computer Society (2017) 3

[12] Camenisch, J., Drijvers, M., Lehmann, A.: Anonymous attestation using the strong diffie hellman assumption revisited. In: TRUST. Lecture Notes in Computer Science, Springer (2016) 1, 2

[13] Camenisch, J., Dubovitskaya, M., Haralambiev, K., Kohlweiss, M.: Composable and modular anonymous credentials: Definitions and practical constructions. In: ASIACRYPT (2). Lecture Notes in Computer Science, Springer (2015) 3

[14] Camenisch, J., Herreweghen, E.V.: Design and implementation of the *idemix* anonymous credential system. In: CCS. pp. 21–30. ACM (2002) 3

[15] Camenisch, J., Hohenberger, S., Pedersen, M.Ø.: Batch verification of short signatures. J. Cryptol. (2012) 18

[16] Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: EUROCRYPT. Lecture Notes in Computer Science, Springer (2001) 1

[17] Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: SCN. Lecture Notes in Computer Science, Springer (2002) 1

[18] Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: CRYPTO. Lecture Notes in Computer Science, Springer (2004) 1

[19] Cantor, D.G., Zassenhaus, H.: A new algorithm for factoring polynomials over finite fields. Mathematics of Computation (1981). https://doi.org/10.1090/S0025-5718-1981-0606517-5 25, 28, 32, 34

[20] Chairattana-Apirom, R., Hofheinz, D., Tessaro, S.: Tight security for BBS signatures. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 16541, pp. 246–276. Springer (2026) 2, 4

[21] Chaum, D.: Blind signature system. In: CRYPTO. Plenum Press, New York (1983) 10

[22] Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM (1985) 1

[23] Connolly, A., Lafourcade, P., Perez-Kempner, O.: Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. In: Public Key Cryptography (1). Lecture Notes in Computer Science, Springer (2022) 3, 13

[24] Crites, E.C., Kohlweiss, M., Preneel, B., Sedaghat, M., Slamanig, D.: Threshold structure-preserving signatures. In: ASIACRYPT (2). Lecture Notes in Computer Science, Springer (2023) 3

[25] Curran, S., Philipp, A., Yildiz, H., Curren, S., Martinez Jurado, V., Bhaduri, A., Ivanov, A.: AnonCreds specification v1.0 draft. https://anoncreds.github.io/anoncreds-spec/ (2023) 3

[26] Doerner, J., Kondi, Y., Lee, E., Shelat, A., Tyner, L.: Threshold BBS+ signatures for distributed anonymous credential issuance. In: SP. IEEE (2023) 2, 4

[27] European Commission: European digital identity wallet architecture and reference framework. https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework (Feb 2023), published on 10 February 2023. Last updated on 25 August 2023. 1

[28] European Parliament and Council of the European Union: Regulation (EU) 2024/1183 of 11 april 2024 amending regulation (EU) no 910/2014 as regards establishing the european digital identity framework (Apr 2024), official Journal of the European Union, L 202401183. ELI: http://data.europa.eu/eli/reg/2024/1183/oj 1

[29] Faust, S., Hazay, C., Kretzler, D., Rometsch, L., Schlosser, B.: Non-interactive threshold BBS+ from pseudorandom correlations. In: CT-RSA. Lecture Notes in Computer Science, Springer (2025) 2, 4

[30] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: CRYPTO. pp. 186–194. Lecture Notes in Computer Science, Springer (1986) 2

[31] Frigo, M., Shelat, A.: Anonymous credentials from ECDSA. IACR Commun. Cryptol. **3**(1), 7 (2026) 3

[32] Fuchsbauer, G., Hanser, C., Slamanig, D.: Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. J. Cryptol. pp. 498–546 (2019) 3, 4, 20

[33] Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: CRYPTO (2). pp. 33–62. Lecture Notes in Computer Science, Springer (2018) 10, 13

[34] Fuchsbauer, G., Regen, F., Wee, H.: Round-optimal threshold blind signatures without random oracles. IACR Cryptol. ePrint Arch. (2026) 3, 4, 5, 10, 27, 31, 36

[35] Garg, P., Garg, S., Policharla, G., Roberts, B.: Threshold anonymous credentials with silent setup. IACR Cryptol. ePrint Arch. (2025) 2, 3, 4

[36] Hanser, C., Slamanig, D.: Structure-preserving signatures on equivalence classes and their application to anonymous credentials. In: ASIACRYPT (1). Lecture Notes in Computer Science, Springer (2014) 3

[37] Heng, Y., Liu, M., Lu, X., Xue, H., Bao, Z., Au, M.H.: Efficient construction of threshold BBS+ signatures and its extensions. In: PKC (3). Lecture Notes in Computer Science, vol. 16553, pp. 496–527. Springer (2026) 2

[38] Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: ASIACRYPT. Lecture Notes in Computer Science, Springer (2010) 4

[39] Kaviani, D., Setty, S.: Vega: Low-latency zero-knowledge proofs over existing credentials. In: SP. pp. 1951–1969. IEEE (2026) 3

[40] Lehmann, A., Nazarian, P., Özbay, C.: Stronger security for threshold blind signatures. In: EUROCRYPT (2). Lecture Notes in Computer Science, vol. 15602, pp. 335–364. Springer (2025) 12

[41] Looker, T., Kalos, V., Whitehead, A., Lodder, M.: The BBS signature scheme. Internet-Draft draft-irtf-cfrg-bbs-signatures-10, Internet Research Task Force (Jan 2026), https://datatracker.ietf.org/doc/draft-irtf-cfrg-bbs-signatures/, work in Progress 1

[42] Mir, O., Bauer, B., Griffy, S., Lysyanskaya, A., Slamanig, D.: Aggregate signatures with versatile randomization and issuer-hiding multi-authority anonymous credentials. In: CCS. ACM (2023) 13

[43] Mitrokotsa, A., Mukherjee, S., Sedaghat, M., Slamanig, D., Tomy, J.: Threshold structure-preserving signatures: Strong and adaptive security under standard assumptions. In: Public Key Cryptography (1). Lecture Notes in Computer Science, Springer (2024) 3

[44] Paquin, C., Policharla, G., Zaverucha, G.: Crescent: Stronger privacy for existing credentials. IACR Cryptol. ePrint Arch. **2024**, 2013 (2024) 3

[45] Paquin, C., Zaverucha, G.: U-prove cryptographic specification v1. 1. Technical Report, Microsoft Corporation (2011) 3

[46] Pointcheval, D., Sanders, O.: Short randomizable signatures. In: CT-RSA. pp. 111–126. Lecture Notes in Computer Science, Springer (2016) 3, 4, 20

[47] Rometsch, L., Lehwalder, P., Faust, S., Schulte, S.: Delegatable anonymous credentials from legacy credentials using recursive zk-snarks. IACR Cryptol. ePrint Arch. p. 1855 (2026) 3

[48] Rosenberg, M., White, J.D., Garman, C., Miers, I.: zk-creds: Flexible anonymous credentials from zksnarks and existing identity infrastructure. In: SP. pp. 790–808. IEEE (2023) 3

[49] Sanders, O.: Efficient redactable signature and application to anonymous credentials. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 12111, pp. 628–656. Springer (2020) 3, 4

[50] Shamir, A.: How to share a secret. Commun. ACM (1979) 9

[51] Sonnino, A., Al-Bassam, M., Bano, S., Meiklejohn, S., Danezis, G.: Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers. In: NDSS. The Internet Society (2019) 3, 4

[52] Tan, S., Groß, T.: Monipoly - an expressive q-sdh-based anonymous attribute-based credential system. In: ASIACRYPT (3). pp. 498–526. Lecture Notes in Computer Science, Springer (2020) 2, 4, 20

[53] Tessaro, S., Zhu, C.: Revisiting BBS signatures. In: EUROCRYPT (5). Lecture Notes in Computer Science, Springer (2023) 1, 2, 4, 20

A Extended Proofs of BB-KZG and Blind-BB-KZG

In this section, we will provide complete security proofs for our signature and blind signature constructions.

A.1 Security Proof of BB-KZG

Theorem 5.1 (restated). *If the bilinear* q-DL *assumption holds,* BB-KZG *is* tEUF-CMA *secure as per Definition 3.4 in the algebraic group model.*

Proof. The proof proceeds in a similar way as the proof of Theorem 7.4, but for slightly different group elements. In BB-KZG, the adversary can query $\mathcal{O}^{\mathsf{PSign}}$ on a set of attributes instead of a polynomial and outputs the attribute set alongside the forgery directly. We construct a reduction to q-DL that uses the coefficients output by an adversary $\mathcal{A}$ for tEUF-CMA in the algebraic group model. We prove that for all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$, such that for all $\lambda, n, l \in \mathbb{N}$, it holds that

$$\mathsf{Adv}^{\mathsf{tEUF\text{-}CMA}}_{\mathcal{A},\text{BB-KZG}}(\lambda, n, l) \leq \mathsf{Adv}^{\mathsf{q\text{-}DL}}_{\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda),$$

where tEUF-CMA is defined in Definition 3.4.

Let q_I be the number of partial issuance queries made by $\mathcal{A}$. In tEUF-CMA, $\mathcal{A}$'s forgery output is $(\sigma_1, \sigma_2, M^*)$, where M^* denotes the attribute set for which σ_1 and σ_2 are valid. In the AGM, $\mathcal{A}$ additionally outputs coefficient vectors $\mathbf{v}$ and $\mathbf{v}'$ that describe the two group elements σ_1 and σ_2 as linear combinations of previously seen group elements $S \subset \mathbb{G}_1$ presented to $\mathcal{A}$ in Game_1. It holds that

$$S = \left\{ g_1, h_1, \left\{ g_1^{\tau^i} \right\}_{i\in[l]}, \left\{ h_1^{\tau^i} \right\}_{i\in[l]}, \left\{ g_1^{\mathsf{sk}_{\mathcal{I}(i)}} h_1^{r_i p_{M_i}(\tau)}, g_1^{r_i} \right\}_{i\in[q_I]}, \left\{ g_1^{\mathsf{sk}_c} \right\}_{c\in\mathcal{C}} \right\},$$

where $\mathcal{I}(x) : [q_I] \to [n]$ maps a queried signature to its signer. For $\mathcal{A}$ to win the game, it must hold that $\mathbf{e}(\sigma_1, g_2) = \mathbf{e}(\sigma_2, h_2^{p_{M^*}(\tau)})$. The pairing check implies the following polynomial equation in the exponent:

$$\begin{aligned} f(\{\mathsf{sk}_i\}_{i\in[n]}, \nu, \tau, \{r_i\}_{i\in[q_I]}) &= v_0 + v_1\nu + \sum_{i\in[l]} \mathbf{v}_2[i]\tau^i + \sum_{i\in[l]} \mathbf{v}_3[i]\nu\tau^i + \sum_{i\in[q_I]} \mathbf{v}_4[i](\mathsf{sk}_{\mathcal{I}(i)} + \nu r_i p_{M_i}(\tau)) \\ &+ \mathbf{v}_5[i]r_i + \sum_{c\in\mathcal{C}} \mathbf{v}_6[c]\mathsf{sk}_c - \mathsf{sk} - \nu p_{M^*}(\tau) \Bigg(v_0' + v_1'\nu + \sum_{i\in[l]} \mathbf{v}_2'[i]\tau^i + \sum_{i\in[l]} \mathbf{v}_3'[i]\nu\tau^i \\ &+ \sum_{i\in[q_I]} \mathbf{v}_4'[i](\mathsf{sk}_{\mathcal{I}(i)} + \nu r_i p_{M_i}(\tau)) + \mathbf{v}_5'[i]r_i + \sum_{c\in\mathcal{C}} \mathbf{v}_6'[c]\mathsf{sk}_c \Bigg). \end{aligned}$$

$\mathcal{A}$ is successful in tEUF-CMA, iff f evaluates to zero for the specific choice of exponents defined in the game. Let $\text{Game}_0 = \mathsf{tEUF\text{-}CMA}$ in the following.

Game_1: We define Game_1 to be equal to Game_0, but an algebraic adversary $\mathcal{A}$ wins only if polynomial f, defined as above, is not the zero polynomial. Assume f is 0. Then, the adversarial strategy yields a valid forgery for any choice of setup, keys, and signature randomness. All unique products of variables must have a coefficient of 0, implying that $\mathbf{v}_4' = \vec{0}$, as these terms are the only ones to contain $\nu^2 r_i$. Further, all r_i terms must cancel, implying that $\mathbf{v}_5 = \vec{0}$. $\mathsf{sk}_c\nu$ terms are unique, implying that $\mathbf{v}_6' = \vec{0}$. To cancel νr_i, it must hold $\forall i \in [q_I]$:

$$\mathbf{v}_4[i]\nu p_{M_i}(\tau) - \mathbf{v}_5'[i]\nu p_{M^*}(\tau) = 0.$$

This holds if $\mathbf{v}_4[i] = \mathbf{v}_5'[i] = 0$, or if $p_{M^*} | p_{M_i}$. For all $i \in [q_I]$ for which p_{M^*} does not divide p_{M_i}, it follows that $\mathbf{v}_4[i] = \mathbf{v}_5'[i] = 0$. If p_{M^*} divides p_{M_i}, it holds that $M^* \subseteq M_i$. By the game's win condition, it then further holds that $|\mathcal{C} \cup \mathcal{U}_{\mathsf{iss}}[M_i]| < t$. Therefore, $\mathbf{v}_4$ has less than $t - |\mathcal{C}|$ entries that are non-zero. Then, by the security of Shamir sharing, $\sum_{c\in\mathcal{C}} \mathbf{v}_6[c]\mathsf{sk}_c + \sum_{i\in[q_I]} \mathbf{v}_4[i]\mathsf{sk}_{\mathcal{I}(i)}$ is independent of sk. However, the polynomial still contains a constant (negated) sk term. This term cannot be cancelled, posing a contradiction to the assumption that f is the zero polynomial. Therefore, any successful adversary $\mathcal{A}$ in Game_0 must output coefficients whose corresponding polynomial f is non-zero. It follows that

$$\mathsf{Adv}^{\text{Game}_0}_{\mathcal{A},\text{BB-KZG}}(\lambda, n, l) = \mathsf{Adv}^{\text{Game}_1}_{\mathcal{A},\text{BB-KZG}}(\lambda, n, l).$$

We now consider a reduction $\mathcal{B}$ for q-DL that receives as input a challenge vector $(\mathsf{BG}, \{g_1^{z^i}, g_2^{z^i}\}_{i\in[l+3]})$ (i.e. $q = l+3$). As in Lemma 3.1, $\mathcal{B}$ embeds the challenge z into every variable x of f by setting $x := \tilde{x}z+\bar{x}$, where the pair $\tilde{x}, \bar{x} \xleftarrow{\$} \mathbb{F}_p$ is sampled independently for each variable. In detail, $\mathcal{B}$ samples $\tilde{\tau}, \bar{\tau}, \tilde{\nu}, \bar{\nu} \xleftarrow{\$} \mathbb{F}_p$ and outputs the BB-KZG public parameters as

$$\mathsf{pp} = \left(\mathsf{BG}, g_1^{\tilde{\nu}z+\bar{\nu}}, g_2^{\tilde{\nu}z+\bar{\nu}}, \{g_1^{(\tilde{\tau}z+\bar{\tau})^i}, g_1^{(\tilde{\nu}z+\bar{\nu})\cdot(\tilde{\tau}z+\bar{\tau})^i}, g_2^{(\tilde{\nu}z+\bar{\nu})\cdot(\tilde{\tau}z+\bar{\tau})^i}\}_{i\in[l]}\right).$$

By doing so, $\mathcal{B}$ embeds the randomized challenge in the exponent of the h generators. For key generation, $\mathcal{B}$ samples $\tilde{\mathsf{sk}}_i, \bar{\mathsf{sk}}_i \xleftarrow{\$} \mathbb{F}_p$, $\forall i \in [n]$, and $\tilde{\mathsf{sk}}, \bar{\mathsf{sk}} \xleftarrow{\$} \mathbb{F}_p$. It then simulates Shamir secret sharing by choosing a function $F(Z) := (\tilde{\mathsf{sk}}z + \bar{\mathsf{sk}}) + \sum_{i=1}^{t-1}(\tilde{\mathsf{sk}}_i z + \bar{\mathsf{sk}}_i)Z^i$, chooses $\mathsf{sk}_i = F(i)$ and sets $g_1^{\mathsf{sk}_i}$ as the secret key of party i. The simulated shares have the same structure as regular Shamir secret shares. Since F is a polynomial of degree $t-1$, up to $t-1$ shares (function values) do not allow an adversary to learn anything about the secret. Also, for $I \subseteq [n]$, $|I| \geq t$, it holds that $(\tilde{\mathsf{sk}}z + \bar{\mathsf{sk}}) = \sum_{i\in I} \lambda_i F(i)$. $\mathcal{B}$ then sets $\mathsf{vk} \leftarrow \mathbf{e}(g_1^{\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}}}, g_2)$.

Upon receiving the ith issuance query on an attribute set M_i, $\mathcal{B}$ samples $\tilde{r}_i, \bar{r}_i \xleftarrow{\$} \mathbb{F}_p$ and uses $\tilde{r}_i z + \bar{r}_i$, a masked q-DL challenge value, for signature generation. $\mathcal{B}$ can compute a valid signature given the q-DL challenge as follows:

$$\sigma_{1,i} \leftarrow g_1^{\mathsf{sk}_{\mathcal{I}(i)}} \cdot g_1^{p_{M_i}(\tilde{\tau}z+\bar{\tau})\cdot(\tilde{r}_i z+\bar{r}_i)\cdot(\tilde{\nu}z+\bar{\nu})}, \qquad \sigma_{2,i} \leftarrow g_1^{\tilde{r}_i z+\bar{r}_i}.$$

$\mathcal{B}$ can easily answer corruption queries by revealing $g_1^{\mathsf{sk}_i}$. Since this term is computable from the q-DL challenge for any $i \in [n]$ and was used to compute $\sigma_{1,i}$, $\mathcal{B}$ perfectly simulates a run of $\textsc{Game}_1$ to $\mathcal{A}$.

If $\mathcal{A}$ outputs a valid forgery, the forgery coefficients define a polynomial f' in z of degree at most $2l+3$. While f is nonzero by the abort condition in $\textsc{Game}_1$, f' might be zero with a small probability, even though f is not. The following holds due to Lemma 3.1:

1. Since f has degree larger than 0, f' must be of the same degree.
2. The coefficient of f''s highest degree monomial is a polynomial in the $\tilde{x}$ values, that is, in $\tilde{\tau}, \tilde{\nu}, \tilde{\mathsf{sk}}$, $\{\tilde{\mathsf{sk}}_i\}_{i\in[t-1]}$ and $\{\tilde{r}_i\}_{i\in[q_I]}$.
3. The $\tilde{x}$ blinding terms hide z from $\mathcal{A}$'s view for each variable x, while $\bar{x}$ perfectly hides $\tilde{x}$.

We can therefore upper bound the probability that f' is zero using the Schwartz-Zippel Lemma by $\frac{2l+4}{p}$, where p is the order of the field, knowing that f' is of degree $2l+3$. $\mathcal{B}$ can solve the polynomial equation implied by $\mathcal{A}$ for z in polynomial time using standard techniques (e.g. the Cantor-Zassenhaus algorithm [19]). It follows overall that

$$\mathsf{Adv}^{\textsc{Game}_0}_{\mathcal{A},\text{BB-KZG}}(\lambda, n, l) = \mathsf{Adv}^{\textsc{Game}_1}_{\mathcal{A},\text{BB-KZG}}(\lambda, n, l) \leq \mathsf{Adv}^{\text{q-DL}}_{\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda),$$

completing the result. □

A.2 Security Proofs of BLIND-BB-KZG

Theorem 6.1 (restated). BLIND-BB-KZG *fulfills blindness as per Definition 3.7.*

We prove blindness in a similar fashion as unlinkability for BBCRED (Theorem 7.2), but without considering the presentation protocol.

Proof. We show blindness against an unbounded adversary. To do so, we argue that the distribution of a partial blind signing query is independent of the underlying signature.

In tBLIND(Figure 2), an adversary $\mathcal{A}$ participates in the partial blind signing procedure for two signatures, σ_0 and σ_1, on M_0 and M_1, respectively. M_0 and M_1 are chosen by the adversary and both queries reveal $\mathsf{Rev} \subseteq M_0 \cap M_1$. Let $p_{M_b} = p_{\mathsf{Hid}_b} \cdot p_{\mathsf{Rev}}$ denote the polynomial whose roots are the attributes in M_b (as defined in the construction), where p_{Hid_b} and p_{Rev} are the polynomials representing the hidden and revealed attributes for $b \in \{0,1\}$, respectively. The challenger samples $b \xleftarrow{\$} \{0,1\}$ and flips the order of running the issuance queries depending on the outcome.

We next argue that each blind signing query will be generated from M_0 and M_1 with the same probability. We define the two terms of the query as a function of the randomness a.

$$C_b(a) \leftarrow h_1^{a p_{M_b}(\tau)}, \quad \hat{C}_b(a) \leftarrow g_2^{a p_{b,\mathsf{Hid}}(\tau)},$$

If $\tau \in M_0 \cup M_1$, the user aborts and the game outputs 0. Assume $p_{M_0}(\tau), p_{M_1}(\tau) \neq 0$.

Fix, without loss of generality, a query for $b = 0$ with randomness a. We next show that there exists exactly one a' that yields the same query for M_1, and vice versa. To this end, we define a bijection $f : a \mapsto a'$ on the query randomness. Define

$$f(a) = a\frac{p_{M_0}(\tau)}{p_{M_1}(\tau)} = a'$$

Since this function must have a slope that is unequal to 0 this map is bijective.

Querying M_0 for a and querying M_1 for $f(a) = a'$ yields the same group elements. Using that $\frac{p_{M_0}(\tau)}{p_{M_1}(\tau)} = \frac{p_{0,\mathsf{Hid}}(\tau)}{p_{1,\mathsf{Hid}}(\tau)}$, since $\mathsf{Rev} \subseteq M_0 \cap M_1$, we find that:

$$C_0(a) = h_1^{a p_{M_0}(\tau)} = h_1^{a' \frac{p_{M_1}(\tau)}{p_{M_0}(\tau)} p_{M_0}(\tau)} = C_1(a') = C_1(f(a))$$
$$\hat{C}_0(a) = g_2^{a p_{0,\mathsf{Hid}}(\tau)} = g_2^{a' \frac{p_{1,\mathsf{Hid}}(\tau)}{p_{0,\mathsf{Hid}}(\tau)} p_{0,\mathsf{Hid}}(\tau)} = \hat{C}_1(a') = \hat{C}_1(f(a))$$

Since f is a bijection, this indicates that for every a' there is exactly one unique a. Hence, the distribution of the issuance query is identical for both $b = 0$ and $b = 1$, and the query does not reveal anything about the encoded attributes. □

Theorem 6.2 (restated). *If the bilinear* q-DL *assumption holds,* Blind-BB-KZG *is one-more unforgeable as per Definition 3.6 in the algebraic group model.*

This proof proceeds very similarly to the proof of Theorem 7.4, but does not make use of the presentation extraction lemma. We first extract the polynomial from each issuance query, before reducing to q-DL.

Proof. We show the theorem using the extractability of partial blind issuance queries shown in Lemma 7.1. We then show that the forgeries output by the adversary can be used to break the q-DL assumption in the AGM. In detail, we prove that for all algebraic PPT adversaries $\mathcal{A}$ for $\mathsf{OMUF}_{\text{Blind-BB-KZG}}$, there exists a PPT adversary $\mathcal{C}$ for q-DL, such that for all $\lambda, n, l \in \mathbb{N}$, it holds that

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathcal{A},\text{Blind-BB-KZG}}(\lambda, n, l) \leq \mathsf{Adv}^{\mathsf{q\text{-}DL}}_{\mathcal{C}}(\lambda) + \mathsf{negl}(\lambda),$$

where OMUF is defined in Definition 3.6. Let q_I be the number of partial blind signing queries made by $\mathcal{A}$. We show the statement via a game-hop, before then showing the final reduction to q-DL.

Let $\text{Game}_0 = \mathsf{OMUF}$ in the following.

Game_1: Game_1 is equivalent to Game_0, except for the following change: the partial signing oracle no longer runs the blind signing protocol with the adversary, but receives the polynomial to be signed in the clear (similar to $\mathcal{O}^{\mathsf{PIssue}}$ in tAC-UF). Note that Game_1 is related to Game_0 in the same way as the two variants of tAC-UF described in Theorem 7.3. Similarly to the proof of that theorem, we can build a PPT adversary $\mathcal{B}$ in Game_1, whose success probability differs from that of $\mathcal{A}$ only by a negligible amount. $\mathcal{B}$ simulates Game_0 to $\mathcal{A}$ by relaying all setup parameters, oracle queries and responses to and from its own challenger to $\mathcal{A}$. Whenever $\mathcal{A}$ sends a blind signing query of the form $(C, \hat{C}, \mathsf{Rev})$ that satisfies Equation (4), $\mathcal{B}$ uses the extractor given in Lemma 7.1 to obtain a polynomial p_i, such that $C = h_1^{p_i(\tau)}$ with overwhelming probability, and queries p_i to its own oracle $\mathcal{O}^{\mathsf{PSign}}$. Note that Lemma 7.1 is stated for BBCred, whose adversary obtains a superset of the group elements available here, so the same argument applies with all terms stemming from $\mathcal{O}^{\mathsf{tPresent}}$ omitted. Applying the extractor sequentially to each of the q_I queries, as in the proof of Theorem 7.3, it follows that

$$\mathsf{Adv}^{\text{Game}_0}_{\mathcal{A},\text{Blind-BB-KZG}}(\lambda, n, l) \leq \mathsf{Adv}^{\text{Game}_1}_{\mathcal{B},\text{Blind-BB-KZG}}(\lambda, n, l) + \mathsf{negl}(\lambda).$$

In Game_1, $\mathcal{B}$'s forgery output is $\left(M_j^*, \sigma_j^*\right)_{j\in[k+1]}$ for $k = \lfloor q_I/(t - |\mathcal{C}|)\rfloor$, where the attribute sets M_j^* are pairwise distinct and $\sigma_j^* = (\sigma_{1,j}^*, \sigma_{2,j}^*)$. Fix a $j \in [k+1]$. In the AGM, $\mathcal{B}$ additionally outputs coefficient vectors $\mathbf{v}$ and $\mathbf{v}'$ that describe the two group elements $\sigma_{1,j}^*$ and $\sigma_{2,j}^*$ as linear combinations of previously seen group elements $S \subset \mathbb{G}_1$ presented to $\mathcal{B}$ in Game_1. Since $\mathcal{B}$ only has access to $\mathcal{O}^{\mathsf{PSign}}$ and $\mathcal{O}^{\mathsf{Corr}}$, it holds that

$$S = \left\{ g_1, h_1, \left\{g_1^{\tau^i}\right\}_{i\in[l]}, \left\{h_1^{\tau^i}\right\}_{i\in[l]}, \left\{g_1^{\mathsf{sk}_{\mathcal{I}(i)}} h_1^{r_i p_i(\tau)}, g_1^{r_i}\right\}_{i\in[q_I]}, \left\{g_1^{\mathsf{sk}_c}\right\}_{c\in\mathcal{C}} \right\},$$

where $\mathcal{I}(x) : [q_I] \to [n]$ maps a queried signature to its signer. For the forgery (M_j^*, σ_j^*) to verify, it must hold that $\mathbf{e}(\sigma_{1,j}^*, g_2) = \mathbf{e}(\sigma_{2,j}^*, h_2^{p_{M_j^*}(\tau)})$. The pairing check implies the following polynomial equation in the exponent:

$$f_j(\{\mathsf{sk}_i\}_{i\in[n]}, \nu, \tau, \{r_i\}_{i\in[q_I]}) = v_0 + v_1\nu + \sum_{i\in[l]} \mathbf{v}_2[i]\tau^i + \sum_{i\in[l]} \mathbf{v}_3[i]\nu\tau^i + \sum_{i\in[q_I]} \mathbf{v}_4[i](\mathsf{sk}_{\mathcal{I}(i)} + \nu r_i p_i(\tau))$$
$$+ \mathbf{v}_5[i]r_i + \sum_{c\in\mathcal{C}} \mathbf{v}_6[c]\mathsf{sk}_c - \mathsf{sk} - \nu p_{M_j^*}(\tau) \Bigg(v_0' + v_1'\nu + \sum_{i\in[l]} \mathbf{v}_2'[i]\tau^i + \sum_{i\in[l]} \mathbf{v}_3'[i]\nu\tau^i$$
$$+ \sum_{i\in[q_I]} \mathbf{v}_4'[i](\mathsf{sk}_{\mathcal{I}(i)} + \nu r_i p_i(\tau)) + \mathbf{v}_5'[i]r_i + \sum_{c\in\mathcal{C}} \mathbf{v}_6'[c]\mathsf{sk}_c \Bigg).$$

The forgery (M_j^*, σ_j^*) verifies, iff f_j evaluates to zero for the specific choice of exponents defined in the game.

GAME$_2$: We define GAME$_2$ to be equal to GAME$_1$, but an algebraic adversary $\mathcal{B}$ wins only if at least one of the polynomials f_j, $j \in [k+1]$, defined as above, is not the zero polynomial. Assume all f_j are 0. Then, the adversarial strategy yields a valid forgery for any choice of setup, keys, and signature randomness. All unique products of variables must have a coefficient of 0, implying that $\mathbf{v}_4' = \vec{0}$, as these terms are the only ones to contain $\nu^2 r_i$. Further, all r_i terms must cancel, implying that $\mathbf{v}_5 = \vec{0}$. $\mathsf{sk}_c\nu$ terms are unique, implying that $\mathbf{v}_6' = \vec{0}$. To cancel νr_i, it must hold $\forall i \in [q_I]$:

$$\mathbf{v}_4[i]\nu p_i(\tau) - \mathbf{v}_5'[i]\nu p_{M_j^*}(\tau) = 0.$$

This holds if $\mathbf{v}_4[i] = \mathbf{v}_5'[i] = 0$, or if $p_{M_j^*} | p_i$. For all $i \in [q_I]$ for which $p_{M_j^*}$ does not divide p_i, it follows that $\mathbf{v}_4[i] = \mathbf{v}_5'[i] = 0$. If $p_{M_j^*}$ divides p_i, it holds that $M_j^* \subseteq M_i$, where M_i denotes the roots of p_i. By the win condition of OMUF, the $k+1$ forgeries are on pairwise distinct attribute sets, while $\mathcal{A}$ makes q_I partial signing queries in total. As a signature on a given attribute set requires at least $t - |\mathcal{C}|$ partial signatures from honest signers, and $k = \lfloor q_I/(t - |\mathcal{C}|)\rfloor$, there must be an index $j^* \in [k+1]$ for which $\mathcal{A}$ has obtained less than $t - |\mathcal{C}|$ partial signatures, meaning that $|\mathcal{C} \cup \mathcal{U}_{\mathsf{iss}}[M_{j^*}^*]| < t$. We consider the polynomial f_{j^*} of this forgery index in the following. In f_{j^*}, $\mathbf{v}_4$ has less than $t - |\mathcal{C}|$ entries that are non-zero. By the security of Shamir sharing, $\sum_{c\in\mathcal{C}} \mathbf{v}_6[c]\mathsf{sk}_c + \sum_{i\in[q_I]} \mathbf{v}_4[i]\mathsf{sk}_{\mathcal{I}(i)}$ is then independent of sk. However, the polynomial still contains a constant (negated) sk term. This term cannot be cancelled, posing a contradiction to the assumption that f_{j^*} is the zero polynomial. Therefore, any successful adversary $\mathcal{B}$ in GAME$_1$ must output coefficients whose corresponding polynomial f_{j^*} is non-zero. It follows that

$$\mathsf{Adv}^{\text{GAME}_1}_{\mathcal{B},\text{BLIND-BB-KZG}}(\lambda, n, l) = \mathsf{Adv}^{\text{GAME}_2}_{\mathcal{B},\text{BLIND-BB-KZG}}(\lambda, n, l).$$

We now consider a reduction $\mathcal{C}$ for q-DL that receives as input a challenge vector $(\mathsf{BG}, \{g_1^{z^i}, g_2^{z^i}\}_{i\in[l+3]})$ (i.e. $q = l + 3$). As in the proof of Theorem 7.4, $\mathcal{C}$ embeds the challenge z into every variable x of f_{j^*} by setting $x := \tilde{x}z + \bar{x}$, where the pair $\tilde{x}, \bar{x} \xleftarrow{\$} \mathbb{F}_p$ is sampled independently for each variable. The same blinding technique is used by Fuchsbauer et al. [34]. $\mathcal{C}$ additionally samples $\tilde{\tau}, \bar{\tau}, \tilde{\nu}, \bar{\nu} \xleftarrow{\$} \mathbb{F}_p$ and outputs the BLIND-BB-KZG public parameters as

$$\mathsf{pp} = \Big(\mathsf{BG}, g_1^{\tilde{\nu}z+\bar{\nu}}, g_2^{\tilde{\nu}z+\bar{\nu}}, \{g_1^{(\tilde{\tau}z+\bar{\tau})^i}, g_2^{(\tilde{\tau}z+\bar{\tau})^i}\}_{i\in[l]}, \{g_1^{(\tilde{\nu}z+\bar{\nu})\cdot(\tilde{\tau}z+\bar{\tau})^i}, g_2^{(\tilde{\nu}z+\bar{\nu})\cdot(\tilde{\tau}z+\bar{\tau})^i}\}_{i\in[l]}\Big).$$

By doing so, $\mathcal{C}$ embeds the randomized challenge in the exponent of the h generators. For key generation, $\mathcal{C}$ samples $\tilde{\mathsf{sk}}_i, \bar{\mathsf{sk}}_i \xleftarrow{\$} \mathbb{F}_p$, $\forall i \in [n]$, and $\tilde{\mathsf{sk}}, \bar{\mathsf{sk}} \xleftarrow{\$} \mathbb{F}_p$. It then simulates Shamir secret sharing by choosing a function $F(Z) := (\tilde{\mathsf{sk}}z + \bar{\mathsf{sk}}) + \sum_{i=1}^{t-1}(\tilde{\mathsf{sk}}_i z + \bar{\mathsf{sk}}_i)Z^i$, chooses $\mathsf{sk}_i = F(i)$ and sets $g_1^{\mathsf{sk}_i}$ as the secret key of party i. The simulated shares have the same structure as regular Shamir secret shares. Since F is a polynomial of degree $t-1$, up to $t-1$ shares (function values) do not allow an adversary to learn anything about the secret. Also, for $I \subseteq [n]$, $|I| \geq t$, it holds that $(\tilde{\mathsf{sk}}z + \bar{\mathsf{sk}}) = \sum_{i\in I} \lambda_i F(i)$. $\mathcal{C}$ then sets $\mathsf{vk} \leftarrow \mathbf{e}(g_1^{\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}}}, g_2)$.

Upon receiving the ith partial signing query on a polynomial p_i, $\mathcal{C}$ samples $\tilde{r}_i, \bar{r}_i \xleftarrow{\$} \mathbb{F}_p$ and uses $\tilde{r}_i z + \bar{r}_i$, a masked q-DL challenge value, for signature generation. $\mathcal{C}$ can compute a valid signature given the q-DL challenge as follows:

$$\sigma_{1,i} \leftarrow g_1^{\mathsf{sk}_{\mathcal{I}(i)}} \cdot g_1^{p_i(\tilde{\tau}z+\bar{\tau})\cdot(\tilde{r}_i z+\bar{r}_i)(\tilde{\nu}z+\bar{\nu})}, \qquad \sigma_{2,i} \leftarrow g_1^{\tilde{r}_i z+\bar{r}_i}.$$

$\mathcal{C}$ can easily answer corruption queries by revealing $g_1^{\mathsf{sk}_i}$. Since this term is computable from the q-DL challenge for any $i \in [n]$ and was used to compute $\sigma_{1,i}$, $\mathcal{C}$ perfectly simulates a run of GAME_2 to $\mathcal{B}$.

If $\mathcal{B}$ outputs valid forgeries, the coefficients of the forgery j^* define a polynomial f' in z of degree at most $2l+3$. f_{j^*} is nonzero by the abort condition in GAME_2. With a small probability, f' is equal to the zero polynomial, even though f_{j^*} is not. The following holds due to Lemma 3.1:

1. Since f_{j^*} has degree larger than 0, f' must be of the same degree.
2. The coefficient of f''s highest degree monomial is a polynomial in the $\tilde{x}$ values, that is, in $\tilde{\tau}, \tilde{\nu}, \tilde{\mathsf{sk}}$, $\{\tilde{\mathsf{sk}}_i\}_{i\in[t-1]}$ and $\{\tilde{r}_i\}_{i\in[q_I]}$.
3. The $\tilde{x}$ blinding terms hide z from $\mathcal{B}$'s view for each variable x, while $\bar{x}$ perfectly hides $\tilde{x}$.

We can therefore upper bound the probability that f' is zero using the Schwartz-Zippel Lemma by $\frac{2l+4}{p}$, where p is the order of the field, knowing that f' is of degree $2l+3$. $\mathcal{C}$ can solve the polynomial equation implied by $\mathcal{B}$ for z in polynomial time using standard techniques (e.g. the Cantor-Zassenhaus algorithm [19]). It follows overall that

$$\mathsf{Adv}^{\text{GAME}_0}_{\mathcal{A},\text{BLIND-BB-KZG}}(\lambda,n,l) \leq \mathsf{Adv}^{\text{GAME}_2}_{\mathcal{B},\text{BLIND-BB-KZG}}(\lambda,n,l) + \mathsf{negl}(\lambda) \leq \mathsf{Adv}^{\mathsf{q\text{-}DL}}_{\mathcal{C}}(\lambda) + \mathsf{negl}(\lambda),$$

completing the result. □

B Extended Proofs of BBCred

In this section, we give the complete security proofs for our threshold credential construction BBCred.

B.1 Unlinkability of BBCred

Theorem 7.2 (restated). *Our construction* BBCred *fulfills unlinkability as per Definition 4.4.*

Proof. We next show unlinkability against an unbounded adversary, i.e., even a computationally unbounded adversary cannot link a presentation to the underlying credential or issuance query. To show this, it suffices to argue that the distribution of each partial issuance query is independent of the queried attribute set and the distribution of the presentation is independent of the underlying credential.

The adversary provides two attribute sets, M_0 and M_1, such that $\mathsf{Rev} \subseteq M_0 \cap M_1$. Let $p_{M_b} = p_{\mathsf{Hid}_b} \cdot p_{\mathsf{Rev}}$ denote the polynomial whose roots are the attributes in M_b (as defined in the construction), where p_{Hid_b} and p_{Rev} are the polynomials representing the hidden and revealed attributes for $b \in \{0,1\}$, respectively. The challenger samples $b \xleftarrow{\$} \{0,1\}$, requests a credential σ_b on M_b, and sends a presentation for σ_b.

By Theorem 6.1, a partial issuance query reveals nothing about the attribute set it encodes. Concretely, the map $f(a) = a\frac{p_{M_0}(\tau)}{p_{M_1}(\tau)}$ is a bijection on $\mathbb{F}_p^*$, and querying M_0 with randomness a produces exactly the same group elements as querying M_1 with randomness $f(a)$. As the randomness is drawn uniformly at random, a query is therefore equally likely to encode M_0 or M_1.

Consider now two credentials σ_0 and σ_1 that are the result of holder-local aggregation. Depending on the value of b, the adversary is shown a presentation of either. We next argue that the presentation can be generated from σ_0 and σ_1 with the same probability. According to the construction, the credential σ_b for $b \in \{0,1\}$ is defined as $\sigma_b = (\sigma_{b,1}, \sigma_{b,2})$, where

$$\sigma_{1,b} = g_1^{sk} \cdot C_b^{r_b}, \quad \sigma_{2,b} = g_1^{r_b}, \quad C_b = h_1^{p_{M_b}(\tau)},$$

with $r_b = t_b v_b$ for uniformly random t_b, v_b, where t_b is the blinding randomness and v_b is the signing randomness. Note that in the following argument we do not assume that the adversary does not know r_b. Indeed, the argument still holds even if the adversary knows it (which covers the case of adversarial issuers).

We next define the presentation of σ_b as a function of the randomness a_0, a_1 which is uniformly distributed and sampled by the challenger. By construction, it holds that

$$C_b(a_0,a_1) \leftarrow h_2^{a_1\, p_{M_b}(\tau)}, \qquad \sigma'_{1,b}(a_0,a_1) \leftarrow \sigma_{1,b} \cdot h_1^{a_0\, p_{M_b}(\tau)},$$
$$\hat{C}_{b,\mathsf{Hid}}(a_0,a_1) \leftarrow g_1^{a_1\, p_{b,\mathsf{Hid}}(\tau)(\tau-\mathsf{aux})}, \qquad \sigma'_{2,b}(a_0,a_1) \leftarrow (\sigma_{2,b} \cdot g_1^{a_0})^{1/a_1}.$$

Expanding the presentation terms, we obtain

$$\sigma'_{1,b}(a_0, a_1) = g_1^{sk} h_1^{(r_b + a_0) p_{M_b}(\tau)}, \qquad \sigma'_{2,b}(a_0, a_1) = g_1^{(a_0 + r_b)/a_1}.$$

If $\tau \in M_b$, the holder aborts during issuance and the game outputs 0. By embedding τ in M_0 or M_1, the adversary can win with probability at most $1/2$. Assume now $p_{M_0}(\tau), p_{M_1}(\tau) \neq 0$. Fix, without loss of generality, a presentation for $b = 0$ with randomness (a_0, a_1). We next show that there exists exactly one pair (a'_0, a'_1) that yields the same presentation when using σ_1, and vice versa. To this end, we define a bijection $f : (a_0, a_1) \mapsto (a'_0, a'_1)$ on the presentation randomness. Let

$$\gamma := \frac{p_{M_0}(\tau)}{p_{M_1}(\tau)},$$

and define

$$f(a_0, a_1) = ((r_0 + a_0)\gamma - r_1, a_1\gamma) = (a'_0, a'_1).$$

Since $\gamma \neq 0$, this map is bijective.

We now show that using (a'_0, a'_1) to present σ_1 yields the same presentation. First,

$$\sigma'_{1,1}(a'_0, a'_1) = g_1^{sk} h_1^{(r_1 + a'_0) p_{M_1}(\tau)} = g_1^{sk} h_1^{(r_1 + (r_0 + a_0)\gamma - r_1) p_{M_1}(\tau)}.$$

Substituting $\gamma = p_{M_0}(\tau)/p_{M_1}(\tau)$, we obtain

$$\sigma'_{1,1}(a'_0, a'_1) = g_1^{sk} h_1^{(r_0 + a_0) p_{M_0}(\tau)} = \sigma'_{1,0}(a_0, a_1).$$

Next,

$$\sigma'_{2,1}(a'_0, a'_1) = g_1^{(a'_0 + r_1)/a'_1} = g_1^{((r_0 + a_0)\gamma - r_1 + r_1)/(a_1\gamma)} = g_1^{(r_0 + a_0)/a_1} = \sigma'_{2,0}(a_0, a_1).$$

Moreover,

$$C_1(a'_0, a'_1) = h_2^{a'_1 p_{M_1}(\tau)} = h_2^{a_1 \gamma p_{M_1}(\tau)} = h_2^{a_1 p_{M_0}(\tau)} = C_0(a_0, a_1).$$

Now, by the game, it must hold that $\mathsf{Rev} \subseteq M_0 \cap M_1$, hence, it holds that

$$\gamma = \frac{p_{M_0}(\tau)}{p_{M_1}(\tau)} = \frac{p_{\mathsf{Hid}_0}(\tau)}{p_{\mathsf{Hid}_1}(\tau)}.$$

Therefore,

$$\hat{C}_1(a'_0, a'_1) = g_1^{a'_1 p_{\mathsf{Hid}_1}(\tau)(\tau - \mathsf{aux})} = g_1^{a_1 \gamma p_{\mathsf{Hid}_1}(\tau)(\tau - \mathsf{aux})} = g_1^{a_1 p_{\mathsf{Hid}_0}(\tau)(\tau - \mathsf{aux})} = \hat{C}_0(a_0, a_1).$$

To conclude the proof, since f is a bijection, this indicates that for every (a'_0, a'_1) there is exactly one unique pair (a_0, a_1). Hence, the distribution of the presentation is identical for both $b = 0$ and $b = 1$. Therefore, the presentation does not reveal any information about the underlying credential. □

B.2 Unforgeability of BBCred

We begin by showing that a partial blind issuance oracle $\mathcal{O}^{\mathsf{PBIssue}}$ can be simulated using the $\mathcal{O}^{\mathsf{PIssue}}$ oracle of tAC-UF, which handles the game-specific bookkeeping. As in Section 7, let $\mathcal{O}^{\mathsf{PBIssue}}$ be defined as $\mathcal{O}^{\mathsf{PIssue}}$ (Figure 3), with the distinction that $\mathcal{O}^{\mathsf{PBIssue}}$ does not take the polynomial as input, extract the roots, call PVerify or update $\mathcal{U}_{\mathsf{iss}}$.

Recall that tAC-UF gives $\mathcal{B}$ access to a partial issuance oracle, in which the issuer learns the polynomial which $\mathcal{B}$ wants to get signed. In contrast, the $\mathcal{O}^{\mathsf{PBIssue}}$ oracle hides the polynomial to be signed. The proof is given in the following, and uses Lemma 7.1. The lemma shows that, given a valid issuance query, i.e., one that satisfies Equation (4), under the q-DL assumption, there exists a straight-line extractor in the AGM that can recover the encoded polynomial. By recursively applying the extractor, we show that $\mathcal{O}^{\mathsf{PBIssue}}$ can be simulated using $\mathcal{O}^{\mathsf{PIssue}}$.

Theorem 7.3 (restated). *Let $\mathcal{A}$ be an algebraic PPT adversary in Game* tAC-UF *(Figure 3) with access to $\mathcal{O}^{\mathsf{PBIssue}}$ in place of $\mathcal{O}^{\mathsf{PIssue}}$. Let $(C_k, \hat{C}_k, \mathsf{Rev}_k)$ be the message of $\mathcal{A}$ in its k-th non-aborting query to $\mathcal{O}^{\mathsf{PBIssue}}(i_k)$. Under the bilinear* q-DL *assumption, there exists a PPT adversary $\mathcal{B}$ for* tAC-UF *that runs $\mathcal{A}$ such that, except with negligible probability,*

1. *$\mathcal{B}$ perfectly simulates the view of $\mathcal{A}$, and*

2. *$\mathcal{B}$ queries $\mathcal{O}^{\mathsf{PIssue}}$ exactly on the pairs (p_k, i_k), where $C_k = h_1^{p_k(\tau)}$ and $p_{\mathsf{Rev}_k} \mid p_k$.*

Proof. We use the map $\mathcal{I} : [q_I] \to [n]$ to denote the issuer of the i-th partial issuance query. We inductively show that $\mathcal{B}$ can simulate an interaction with the partial blind issuance oracle $\mathcal{O}^{\mathsf{PBIssue}}$, given only access to an $\mathcal{O}^{\mathsf{PIssue}}$ oracle. $\mathcal{B}$ relays all setup and query information to $\mathcal{A}$ and by the following argument can extract from $\mathcal{A}$'s $\mathcal{O}^{\mathsf{PBIssue}}$ oracle queries:

Base Case: Assume $\mathcal{A}$ sends its first non-aborting query of the form $(C, \hat{C}, \mathsf{Rev})$ to a blind issuance oracle $\mathcal{O}^{\mathsf{PBIssue}}$. $\mathcal{A}$ can be seen as an adversary with access to $\mathcal{O}^{\mathsf{PIssue}}$ that has not made a query yet. As shown in Lemma 7.1, if the q-DL assumption holds, $\mathcal{B}$ can extract a polynomial p, such that $C = h_1^{p(\tau)}$ and $p_{\mathsf{Rev}} \mid p$, and can compute the $\mathcal{O}^{\mathsf{PBIssue}}$ response using its access to $\mathcal{O}^{\mathsf{PIssue}}$.

Inductive Step: For any $q_I + 1 \in \mathbb{N}$, if all previous issuance queries were of the form $C \leftarrow h_1^{p_i(\tau)}$, adversary $\mathcal{A}$ has effectively only had access to oracles $\mathcal{O}^{\mathsf{PIssue}}, \mathcal{O}^{\mathsf{tPresent}}, \mathcal{O}^{\mathsf{Corr}}$. If $\mathcal{A}$ again outputs a non-aborting query $(C, \hat{C}, \mathsf{Rev})$, by Lemma 7.1, assuming the q-DL assumption holds, there exists an efficient extractor that outputs p, such that $C = h_1^{p(\tau)}$ and $p_{\mathsf{Rev}} \mid p$. $\mathcal{B}$ can query p to its $\mathcal{O}^{\mathsf{PIssue}}$ oracle and return the response to $\mathcal{A}$.

If the issuance pairing check fails for a query of $\mathcal{A}$, $\mathcal{B}$ aborts and does not query $\mathcal{O}^{\mathsf{PIssue}}$. By Lemma 7.1, each of the at most q_I extractions fails only with negligible probability. All of $\mathcal{B}$'s responses are distributed exactly as in $\mathcal{O}^{\mathsf{PBIssue}}$, so $\mathcal{B}$ perfectly simulates the view of $\mathcal{A}$, and queries $\mathcal{O}^{\mathsf{PIssue}}$ exactly on the pairs (p_k, i_k). □

Lemma 7.1 (restated). *Let* BGGen *be a bilinear group generator. Let $\mathcal{A}$ be an algebraic PPT adversary that for $\lambda, n, l \in \mathbb{N}$ receives* $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$*, chooses $t \leq n$, and obtains* vk *as input, where* $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$*, and has access to oracles $\mathcal{O}^{\mathsf{PIssue}}, \mathcal{O}^{\mathsf{tPresent}}$ and $\mathcal{O}^{\mathsf{Corr}}$ for* BBCred *as per Definition 4.3. Assuming the bilinear* q-DL *assumption holds, if $\mathcal{A}$ outputs $(C, \hat{C}, \mathsf{Rev})$ such that $\mathbf{e}(C, g_2) = \mathbf{e}(h_1^{p_{\mathsf{Rev}}(\tau)}, \hat{C})$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ with the same input and access to $\mathcal{O}^{\mathsf{PIssue}}$, $\mathcal{O}^{\mathsf{tPresent}}$, $\mathcal{O}^{\mathsf{Corr}}$, and $\mathcal{A}$, that outputs a polynomial p, such that $C = h_1^{p(\tau)}$ and $p_{\mathsf{Rev}} | p$ with probability $1 - \mathsf{negl}(\lambda)$.*

Proof. For convenience, we again define a function $\mathcal{I} : [q_I] \to [n]$, that maps each partial issuance query $i \in [q_I]$ to its signer. We denote a t-out-of-n Shamir secret sharing of a value x by $\mathsf{Shamir.SS}(x, n, t)$ and let $\lambda_{i,n}$ denote the lagrange coefficient of the ith value out of n. The proof proceeds as follows: We give the description of the extractor Ext that uses $\mathcal{A}$'s output to extract the polynomial committed in C. In detail, Ext uses the coefficient representation of C and $\hat{C}$ that $\mathcal{A}$ outputs in the AGM. For Ext to be successful, it must hold that the coefficients output by $\mathcal{A}$ are of a specific form. We show that this is a necessity, assuming that the q-DL assumption holds.

We define the following extractor Ext: Ext forwards the setup pp to $\mathcal{A}$ and returns its choice of the threshold t to its challenger. It forwards the verification keys of all parties to $\mathcal{A}$. Whenever $\mathcal{A}$ queries the $\mathcal{O}^{\mathsf{PIssue}}$, $\mathcal{O}^{\mathsf{tPresent}}$, or $\mathcal{O}^{\mathsf{Corr}}$ oracle, Ext forwards the query to its own oracle and returns the result.

We consider two sets of group elements $S \subset \mathbb{G}_1$ and $S' \subset \mathbb{G}_2$, that $\mathcal{A}$ can obtain from the public parameters, the verification key and its queries to $\mathcal{O}^{\mathsf{PIssue}}$ and $\mathcal{O}^{\mathsf{tPresent}}$. These group elements are parameterized by setup variables $g_1, h_1 = g_1^{\nu}, g_2, h_2 = g_2^{\nu}$, and τ, the secret keys $\mathsf{sk}_{\mathcal{I}(i)}$, $\mathcal{O}^{\mathsf{PIssue}}$ signature randomness r_i and $\mathcal{O}^{\mathsf{tPresent}}$ signature and commitment randomness ρ_j, a_j, and α_j. We define $\beta_j := (\rho_j + a_j)/\alpha_j$ and use β_j in place of ρ_j. As ρ_j is sampled uniformly at random and independently of all other variables, β_j is independent of all other variables, including α_j and a_j. We may thus treat β_j as a fresh variable and substitute $\rho_j + a_j = \alpha_j \beta_j$. The sets are defined as follows:

$$S = \left\{ g_1, h_1, \left\{ g_1^{\tau^i} \right\}_{i\in[l+1]}, \left\{ h_1^{\tau^i} \right\}_{i\in[l]}, \left\{ g_1^{\mathsf{sk}_{\mathcal{I}(i)}} h_1^{r_i p_i(\tau)}, g_1^{r_i} \right\}_{i\in[q_I]}, \left\{ g_1^{\mathsf{sk}_c} \right\}_{c\in\mathcal{C}} \right.$$
$$\left. \left\{ g^{\alpha_j p_{M_{\mathsf{Hid},j}}(\tau)(\tau - \mathsf{aux}_j)}, g_1^{\mathsf{sk}} h_1^{\alpha_j \beta_j p_{M_j}(\tau)}, g_1^{\beta_j} \right\}_{j\in[q_P]} \right\},$$

and

$$S' = \left\{ g_2, h_2, \left\{ g_2^{\tau^i} \right\}_{i\in[l+1]}, \left\{ h_2^{\tau^i} \right\}_{i\in[l]}, \left\{ h_2^{\alpha_j p_{M_j}(\tau)} \right\}_{j\in[q_P]} \right\},$$

where q_I is the number of queries to the partial issuance oracle and q_P is the number of queries to the threshold presentation oracle. $\mathcal{A}$ chooses aux_j, p_i and p_j itself, so these terms are known. In the AGM, $\mathcal{A}$ outputs a coefficient vector alongside each group element, a representation in terms of previously seen group elements. We let $\mathbf{u}$ be the representation of C over S and let $\mathbf{u}'$ be the representation of $\hat{C}$ over S'. Consider the polynomial equation in the exponent induced by the issuance pairing check (Equation (4)), i.e. $\mathbf{e}(C, g_2) = \mathbf{e}\left(h_1^{p_{\mathsf{Rev}}(\tau)}, \hat{C}\right)$. We index vectors $\mathbf{u}$ and $\mathbf{u}'$ to denote coefficients of different elements in S and S'. Using the same variables as S, let f_I be defined as

$$f_I(\{\mathsf{sk}_i\}_{i\notin\mathcal{C}}, \nu, \tau, \{r_i\}_{i\in[q_I]}, \{\alpha_j, \beta_j\}_{j\in[q_P]}) = u_0 + \mathbf{u}_5[i]r_i + u_1\nu + \sum_{i\in[l+1]} \mathbf{u}_2[i]\tau^i + \sum_{i\in[l]} \mathbf{u}_3[i]\nu\tau^i$$
$$+ \sum_{i\in[q_I]} \mathbf{u}_4[i]\left(\mathsf{sk}_{\mathcal{I}(i)} + \nu r_i p_i(\tau)\right) + \sum_{j\in[q_P]} \mathbf{u}_6[j](\mathsf{sk} + \nu\alpha_j\beta_j p_{M_j}(\tau)) + \mathbf{u}_7[j]\beta_j + \mathbf{u}_8[j]\alpha_j p_{\mathsf{Hid}_j}(\tau)(\tau - \mathsf{aux}_j)$$
$$+ \sum_{c\in\mathcal{C}} \mathbf{u}_9[c]\mathsf{sk}_c - \Big(u_0' + u_1'\nu + \sum_{i\in[l+1]} \mathbf{u}_2'[i]\tau^i + \sum_{i\in[l]} \mathbf{u}_3'[i]\nu\tau^i + \sum_{j\in[q_P]} \mathbf{u}_4'[j]\nu\alpha_j p_{M_j}(\tau)\Big)\nu p_{\mathsf{Rev}}(\tau).$$

Note that $f_I(\cdot) = 0$, iff the pairing equation is fulfilled. Consider that for $\mathbf{u}$ and $\mathbf{u}'$, it either holds that f_I is the zero-polynomial or that $f_I(\cdot)$ is zero, but not constant.

Case 1: f_I is not the zero polynomial In this case, one can use algorithm $\mathcal{A}$ to break the q-DL assumption. Consider a reduction $\mathcal{B}$ against q-DL. Given a q-DL challenge in z for $q = l + 3$ of the form $\left(\mathsf{BG}, \left\{g_1^{z^i}, g_2^{z^i}\right\}_{i\in[l+3]}\right)$, $\mathcal{B}$ embeds the challenge z into every variable x of f_I by setting $x := \tilde{x}z + \bar{x}$, where the pair $\tilde{x}, \bar{x} \xleftarrow{\$} \mathbb{F}_p$ is sampled independently for each variable. This is the same blinding technique used by Fuchsbauer et al. [34]. Concretely, $\mathcal{B}$ samples $\tilde{\tau}, \bar{\tau}, \tilde{\nu}, \bar{\nu} \xleftarrow{\$} \mathbb{F}_p$ and outputs public parameters

$$\mathsf{pp} = \left(\mathsf{BG}, g_1^{\tilde{\nu}z+\bar{\nu}}, g_2^{\tilde{\nu}z+\bar{\nu}}, \left\{g_1^{(\tilde{\tau}z+\bar{\tau})^i}, g_2^{(\tilde{\tau}z+\bar{\tau})^i}\right\}_{i\in[l+1]}, \left\{g_1^{(\tilde{\nu}z+\bar{\nu})(\tilde{\tau}z+\bar{\tau})^i}, g_2^{(\tilde{\nu}z+\bar{\nu})(\tilde{\tau}z+\bar{\tau})^i}\right\}_{i\in[l]}\right).$$

For key generation, $\mathcal{B}$ samples $\tilde{\mathsf{sk}}_i, \bar{\mathsf{sk}}_i \xleftarrow{\$} \mathbb{F}_p$, $\forall i \in [n]$, and $\tilde{\mathsf{sk}}, \bar{\mathsf{sk}} \xleftarrow{\$} \mathbb{F}_p$. It then simulates Shamir secret sharing by choosing a function $F(Z) := (\tilde{\mathsf{sk}}z + \bar{\mathsf{sk}}) + \sum_{i=1}^{t-1}(\tilde{\mathsf{sk}}_i z + \bar{\mathsf{sk}}_i)Z^i$, chooses $\mathsf{sk}_i = F(i)$ and sets $g_1^{\mathsf{sk}_i}$ as the secret key of party i.

The simulated shares have the same structure as regular Shamir secret shares. Since F is a polynomial of degree $t-1$, up to $t-1$ shares (function values) do not allow an adversary to learn anything about the secret. Also, for $I \subseteq [n]$, $|I| \geq t$, it holds that $(\tilde{\mathsf{sk}}z + \bar{\mathsf{sk}}) = \sum_{i\in I} \lambda_i F(i)$. $\mathcal{B}$ then sets $\mathsf{vk} \leftarrow \mathbf{e}(g_1^{\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}}}, g_2)$. Again, the additive masking values $\bar{\tau}, \bar{\mathsf{sk}}_i, \bar{\mathsf{sk}}$ and $\bar{\nu}$ perfectly hide the embedded challenge value. To simulate $\mathcal{O}^{\mathsf{PIssue}}$ oracle responses for $\mathcal{A}$, given $M_i, \mathcal{I}(i)$ for $i \in [q_I]$ as input, $\mathcal{B}$ samples random $\tilde{r}_i, \bar{r}_i$ and returns

$$\sigma_{1,j} \leftarrow g_1^{(\tilde{\mathsf{sk}}_{\mathcal{I}(i)}z+\bar{\mathsf{sk}}_{\mathcal{I}(i)})+(\tilde{\nu}z+\bar{\nu})(\tilde{r}_i z+\bar{r}_i)p_i(\tilde{\tau}z+\bar{\tau})}, \quad \sigma_{2,j} \leftarrow g_1^{\tilde{r}_i z+\bar{r}_i}.$$

To simulate $\mathcal{O}^{\mathsf{tPresent}}$, $\mathcal{B}$ samples random $\tilde{\alpha}_j, \bar{\alpha}_j, \tilde{\beta}_j$ and $\bar{\beta}_j$, given M_j, aux_j for $j \in [q_P]$ as input, and computes:

$$C_j \leftarrow g_2^{(\tilde{\nu}z+\bar{\nu})(\tilde{\alpha}_j z+\bar{\alpha}_j)p_{M_j}(\tilde{\tau}z+\bar{\tau})}, \quad \hat{C}_j \leftarrow g_1^{(\tilde{\alpha}_j z+\bar{\alpha}_j)p_{\mathsf{Hid}_j}(\tilde{\tau}z+\bar{\tau})(\tilde{\tau}z+\bar{\tau}-\mathsf{aux}_j)},$$
$$s_{1,j} \leftarrow s g_1^{(\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}})+(\tilde{\alpha}_j z+\bar{\alpha}_j)(\tilde{\nu}z+\bar{\nu})\left(\tilde{\beta}_j z+\bar{\beta}_j\right)p_{M_j}(\tilde{\tau}z+\bar{\tau})}, \quad s_{2,j} \leftarrow g_1^{\tilde{\beta}_j z+\bar{\beta}_j}.$$

Since β_j is sampled directly, no inverse of z has to be embedded. The terms yield a valid and uniformly distributed response to the query, as

$$\mathbf{e}(g_1^{(\tilde{\tau}z+\bar{\tau}-\mathsf{aux}_j)}, C_j) = \mathbf{e}\left(g_1^{(\tilde{\tau}z+\bar{\tau}-\mathsf{aux}_j)(\tilde{\alpha}_j z+\bar{\alpha}_j)p_{\mathsf{Hid}_j}(\tilde{\tau}z+\bar{\tau})}, g_2^{(\tilde{\nu}z+\bar{\nu})p_{\mathsf{Rev}_j}(\tilde{\tau}z+\bar{\tau})}\right) = \mathbf{e}\left(\hat{C}_j, h_2^{p_{\mathsf{Rev}_j}(\tilde{\tau}z+\bar{\tau})}\right),$$

and

$$\mathbf{e}(s_{1,j}, g_2) = \mathbf{e}\left(g_1^{(\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}})+(\tilde{\alpha}_j z+\bar{\alpha}_j)(\tilde{\nu}z+\bar{\nu})\left(\tilde{\beta}_j z+\bar{\beta}_j\right)p_{M_j}(\tilde{\tau}z+\bar{\tau})}, g_2\right)$$
$$= g_T^{\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}}} \cdot \mathbf{e}\left(g_1^{\tilde{\beta}_j z+\bar{\beta}_j}, g_2^{(\tilde{\nu}z+\bar{\nu})(\tilde{\alpha}_j z+\bar{\alpha}_j)p_{M_j}(\tilde{\tau}z+\bar{\tau})}\right) = \mathsf{vk} \cdot \mathbf{e}\left(s_{2,j}, C_j\right).$$

Note that all of the above exponents are polynomials of degree at most $l+3$ in z, so $\mathcal{B}$ can compute them from its q-DL challenge. $\mathcal{B}$ embeds the q-DL challenge in f_I as described in Lemma 3.1. This yields a univariate polynomial f'_I in z. The following holds due to Lemma 3.1:

1. Since f_I has degree larger than 0, f'_I must be of the same degree.
2. The coefficient of f'_I's highest degree monomial is a polynomial in the $\tilde{x}$ values, that is, in $\tilde{\tau}, \tilde{\nu}, \tilde{\mathsf{sk}}$, $\{\tilde{\mathsf{sk}}_i\}_{i\in[t-1]}$, $\{\tilde{r}_i\}_{i\in[q_I]}$ and $\{\tilde{\alpha}_j, \tilde{\beta}_j\}_{j\in[q_P]}$.
3. The $\tilde{x}$ blinding terms hide z from $\mathcal{A}$'s view for each variable x, while $\bar{x}$ perfectly hides $\tilde{x}$.

Because of that, we can upper-bound the probability that f'_I is the zero polynomial even if f_I is not, using the Schwartz-Zippel Lemma. Since f'_I is a polynomial of degree at most $l+3+1+l = 2l+4$, and d is bounded by the same value, f'_I is non-zero with probability at least $1 - \frac{2l+4}{p}$. Using standard techniques [19], $\mathcal{B}$ can then factor f'_I for the challenge z.

Case 2: f_I is the zero polynomial If $f_I(\cdot) = 0$ for any choice of variables, all terms that contain a unique product of variables must have a coefficient of 0, as they cannot be cancelled by a sum of other terms in f_I. We enumerate the terms in the order in which they cancel and list each unique product of variables below.

$$
\begin{aligned}
f_I(\{\mathsf{sk}_i\}_{i\notin\mathcal{C}}, \nu, \tau, \{r_i\}_{i\in[q_I]}, \{\alpha_j, \beta_j\}_{j\in[q_P]}) = {} & \cancel{u_0}^{(7)} + u_1\nu + \sum_{i\in[l+1]} \cancel{\mathbf{u}_2[i]\tau^i}^{(7)} + \sum_{i\in[l]} \mathbf{u}_3[i]\nu\tau^i \\
& + \sum_{i\in[q_I]} \cancel{\mathbf{u}_4[i](\mathsf{sk}_{\mathcal{I}(i)} + \nu r_i p_i(\tau))}^{(8)} + \cancel{\mathbf{u}_5[i] r_i}^{(5)} + \sum_{j\in[q_P]} \cancel{\mathbf{u}_6[j](\mathsf{sk} + \nu\alpha_j\beta_j p_{M_j}(\tau))}^{(2)} + \cancel{\mathbf{u}_7[j]\beta_j}^{(3)} \\
& + \cancel{\mathbf{u}_8[j]\alpha_j p_{M_{\mathsf{Hid},j}}(\tau)(\tau - \mathsf{aux}_j)}^{(4)} + \sum_{c\in\mathcal{C}} \cancel{\mathbf{u}_9[c]\mathsf{sk}_c}^{(9)} - \Bigg(u'_0 + \cancel{u'_1\nu}^{(6)} + \sum_{i\in[l+1]} \mathbf{u}'_2[i]\tau^i \\
& + \cancel{\sum_{i\in[l]} \mathbf{u}'_3[i]\nu\tau^i}^{(6)} + \cancel{\sum_{j\in[q_P]} \mathbf{u}'_4[j]\nu\alpha_j p_{M_j}(\tau)}^{(1)} \Bigg) \nu p_{\mathsf{Rev}}(\tau).
\end{aligned}
$$

The following terms cancel in this equation:

#	Unique Variable Product	#	Unique Variable Product
(1)	$\nu^2\alpha_j$	(5)	r_i
(2)	$\nu\alpha_j\beta_j$	(6)	$\nu^2\tau^x, x \in [0,l]$
(3)	β_j	(7)	$\tau^x, x \in [0,l+1]$
(4)	α_j	(8)	$\nu^2 r_i$
(9)	sk_c		

The remaining polynomial is:

$$
f_I(\cdot) = u_1\nu + \sum_{i\in[l]} \mathbf{u}_3[i]\nu\tau^i - \left(u'_0 + \sum_{i\in[l+1]} \mathbf{u}'_2[i]\tau^i \right) \nu p_{\mathsf{Rev}}(\tau).
$$

The above term is a polynomial only in τ and ν. Given u_1 and $\mathbf{u}_3$, Ext can output a polynomial $p(\tau) = u_1 + \sum_{i\in[l]} \mathbf{u}_3[i]\tau^i$ that fulfills $h_1^{p(\tau)} = C$. As $f_I(\cdot) = 0$, it must further hold that $p_{\mathsf{Rev}} | p$, which we can see by dividing by ν. Only in case 1 is Ext not successful. In this case, we can break the q-DL assumption with probability $1 - \left(\mathsf{Adv}^{\mathsf{q\text{-}DL}}_{\mathcal{B}}(\lambda) + \frac{2l+4}{p}\right)$. Therefore, if the q-DL assumption holds, Ext has a success probability of $1 - \mathsf{negl}(\lambda)$. □

Lemma 7.2 (restated). *Let* BGGen *be a bilinear group generator. Let $\mathcal{A}$ be an algebraic PPT adversary that for $\lambda, n, l \in \mathbb{N}$ receives* $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(\lambda, n, l)$ *and* vk *as input, where* $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[n]}, \mathsf{vk}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pp}, t)$*, and has access to oracles* $\mathcal{O}^{\mathsf{PIssue}}$, $\mathcal{O}^{\mathsf{tPresent}}$, *and* $\mathcal{O}^{\mathsf{Corr}}$ *for* BBCred *as per Definition 4.3. Assuming the bilinear* q-DL *assumption holds, if $\mathcal{A}$ outputs* $(C, \hat{C}, \mathsf{aux}, \mathsf{Rev})$*, such that* $\mathbf{e}\left(g_1^{\tau-\mathsf{aux}}, C\right) = \mathbf{e}\left(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}\right)$ *and* $(\mathsf{Rev}, \mathsf{aux})$ *has not been queried to* $\mathcal{O}^{\mathsf{tPresent}}$*, there exists a PPT extractor* $\mathsf{Ext}_{\mathcal{A}}$ *with*

the same input and access to $\mathcal{O}^{\mathsf{PIssue}}$, $\mathcal{O}^{\mathsf{tPresent}}$ *and* $\mathcal{A}$, *that outputs a polynomial* p, *such that* $C = h_2^{p(\tau)}$ *and* $p_{\mathsf{Rev}}|p$ *with probability* $1 - \mathsf{negl}(\lambda)$.

Proof. The proof proceeds in a similar manner as the proof of Lemma 7.1. However, the input commitments are now in different groups, i.e. $C \in \mathbb{G}_2$ and $\hat{C} \in \mathbb{G}_1$, and we additionally consider auxiliary data aux. Again, let $\mathcal{I}(x) : [q_I] \mapsto [n]$ map to the queried signer for each partial issuance query $i \in [q_I]$. Denote a t-out-of-n Shamir secret sharing of x by $\mathsf{Shamir.SS}(x, n, t)$ and let $\lambda_{i,n}$ be the lagrange coefficient of the ith value out of n. Like in Lemma 7.1, Ext uses $\mathcal{A}$'s output to extract the polynomial committed in C. It does so using the coefficient representation of C and $\hat{C}$ that $\mathcal{A}$ outputs in the AGM. We show that under the $\mathsf{q\text{-}DL}$ assumption, the coefficients output by $\mathcal{A}$ must have a specific form.

On input, Ext forwards the public parameters pp to $\mathcal{A}$ and returns $\mathcal{A}$'s choice of threshold t. It forwards all verification keys to $\mathcal{A}$. Whenever $\mathcal{A}$ queries $\mathcal{O}^{\mathsf{PIssue}}, \mathcal{O}^{\mathsf{tPresent}}$, or $\mathcal{O}^{\mathsf{Corr}}$, Ext forwards the queries and responses to and from its respective oracles. Let the sets of group elements $S \subset \mathbb{G}_1$ and $S' \subset \mathbb{G}_2$ that $\mathcal{A}$ can obtain from interacting with its oracles be defined as in the proof of Lemma 7.1, parameterized by all unknown variables to the adversary. As in the proof of Lemma 7.1, we define $\beta_j := (\rho_j + a_j)/\alpha_j$ for the $\mathcal{O}^{\mathsf{tPresent}}$ randomness. By the same argument, β_j is uniformly distributed and independent of all other variables, so we may treat it as a fresh variable and substitute $\rho_j + a_j = \alpha_j \beta_j$.

As $C \in \mathbb{G}_2$ and $\hat{C} \in \mathbb{G}_1$, we let $\mathbf{u}$ be the representation of C over S' and let $\mathbf{u}'$ be the representation of $\hat{C}$ over S. Consider the polynomial equation implied by the pairing check (Equation (5)), $\mathbf{e}\left(g_1^{\tau - \mathsf{aux}}, C\right) = \mathbf{e}\left(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}\right)$, in the exponent. Define f_P as

$$
\begin{aligned}
f_P(\{\mathsf{sk}_i\}_{i \notin \mathcal{C}}, \nu, \tau, \{r_i\}_{i \in [q_I]}, \{\alpha_j, \beta_j\}_{j \in [q_P]}) = &\left(u_0 + u_1 \nu + \sum_{i \in [l+1]} \mathbf{u}_2[i] \tau^i + \sum_{i \in [l]} \mathbf{u}_3[i] \nu \tau^i \right. \\
&\left. + \sum_{j \in [q_P]} \mathbf{u}_4[j] \nu \alpha_j p_{M_j}(\tau) \right) (\tau - \mathsf{aux}) - \left(u_0' + u_1' \nu + \sum_{i \in [l+1]} \mathbf{u}_2'[i] \tau^i + \sum_{i \in [l]} \mathbf{u}_3'[i] \nu \tau^i \right. \\
&+ \sum_{i \in [q_I]} \mathbf{u}_4'[i](\mathsf{sk}_{\mathcal{I}(i)} + \nu r_i p_i(\tau)) + \mathbf{u}_5'[i] r_i + \sum_{j \in [q_P]} \mathbf{u}_6'[j](\mathsf{sk} + \nu \alpha_j \beta_j p_{M_j}(\tau)) + \mathbf{u}_7'[j] \beta_j \\
&\left. + \mathbf{u}_8'[j] \alpha_j p_{M_{\mathsf{Hid},j}}(\tau)(\tau - \mathsf{aux}_j) + \mathbf{u}_9'[c] \mathsf{sk}_c \right) \nu p_{\mathsf{Rev}}(\tau).
\end{aligned}
$$

Note that $f_P(\cdot) = 0$, iff the pairing equation is fulfilled. Consider that for $\mathbf{u}$ and $\mathbf{u}'$, it either holds that f_P is the zero-polynomial or that $f_P(\cdot)$ is zero, but f_P is not constant.

Case 1: f_P **is not the zero polynomial** In this case, one can use algorithm $\mathcal{A}$ to break the $\mathsf{q\text{-}DL}$ assumption. Consider a reduction $\mathcal{B}$ against $\mathsf{q\text{-}DL}$. Given a $\mathsf{q\text{-}DL}$ challenge in z for $q = l + 3$ of the form $\left(\mathsf{BG}, \left\{g_1^{z^i}, g_2^{z^i}\right\}_{i \in [l+3]}\right)$. As in the proof of Lemma 7.1, $\mathcal{B}$ embeds the challenge z into every variable x of f_P by setting $x := \tilde{x} z + \bar{x}$, where the blinding terms $\tilde{x}, \bar{x} \xleftarrow{\$} \mathbb{F}_p$ is sampled independently for each variable. $\mathcal{B}$ samples $\tilde{\tau}, \bar{\tau}, \tilde{\nu}, \bar{\nu} \xleftarrow{\$} \mathbb{F}_p$ and sets the public parameters to be

$$
\mathsf{pp} = \left(\mathsf{BG}, g_1^{\tilde{\nu} z + \bar{\nu}}, g_2^{\tilde{\nu} z + \bar{\nu}}, \left\{g_1^{(\tilde{\tau} z + \bar{\tau})^i}, g_2^{(\tilde{\tau} z + \bar{\tau})^i}\right\}_{i \in [l+1]}, \left\{g_1^{(\tilde{\nu} z + \bar{\nu})(\tilde{\tau} z + \bar{\tau})^i}, g_2^{(\tilde{\nu} z + \bar{\nu})(\tilde{\tau} z + \bar{\tau})^i}\right\}_{i \in [l]}\right).
$$

For key generation, $\mathcal{B}$ samples $\tilde{\mathsf{sk}}_i, \bar{\mathsf{sk}}_i \xleftarrow{\$} \mathbb{F}_p$, $\forall i \in [n]$, and $\tilde{\mathsf{sk}}, \bar{\mathsf{sk}} \xleftarrow{\$} \mathbb{F}_p$. It then simulates Shamir secret sharing by choosing a function $F(Z) := (\tilde{\mathsf{sk}} z + \bar{\mathsf{sk}}) + \sum_{i=1}^{t-1} (\tilde{\mathsf{sk}}_i z + \bar{\mathsf{sk}}_i) Z^i$, letting $\mathsf{sk}_i = F(i)$ and sets $g_1^{\mathsf{sk}_i}$ as the secret key of party i. The simulated shares have the same structure as regular Shamir secret shares. Since F is a polynomial of degree $t-1$, up to $t-1$ shares (function values) do not allow an adversary to learn anything about the secret. Also, for $I \subseteq [n]$, $|I| \geq t$, it holds that $(\tilde{\mathsf{sk}} z + \bar{\mathsf{sk}}) = \sum_{i \in I} \lambda_i F(i)$. $\mathcal{B}$ then sets sets $\mathsf{vk} \leftarrow \mathbf{e}(g_1^{\tilde{\mathsf{sk}} z + \bar{\mathsf{sk}}}, g_2)$. To simulate $\mathcal{O}^{\mathsf{PIssue}}$ oracle responses for $\mathcal{A}$, given M_i and $\mathcal{I}(i)$ for $i \in [q_I]$ as input, $\mathcal{B}$ samples random $\tilde{r}_i, \bar{r}_i$ and returns

$$
\sigma_{1,j} \leftarrow g_1^{(\tilde{\mathsf{sk}}_{\mathcal{I}(i)} z + \bar{\mathsf{sk}}_{\mathcal{I}(i)}) + (\tilde{\nu} z + \bar{\nu})(\tilde{r}_i z + \bar{r}_i) p_{M_i}(\tilde{\tau} z + \bar{\tau})}, \sigma_{2,j} \leftarrow g_1^{\tilde{r}_i z + \bar{r}_i}.
$$

To simulate $\mathcal{O}^{\mathsf{tPresent}}$, $\mathcal{B}$ samples random $\tilde{a}_j, \bar{a}_j, \tilde{\alpha}_j, \bar{\alpha}_j, \tilde{\rho}_j$ and $\bar{\rho}_j$, given M_j, aux_j for $j \in [q_P]$ as input, and computes:

$$
C_j \leftarrow g_2^{(\tilde{\nu} z + \bar{\nu})(\tilde{\alpha}_j z + \bar{\alpha}_j) p_{M_j}(\tilde{\tau} z + \bar{\tau})}, \quad \hat{C}_j \leftarrow g_1^{(\tilde{\alpha}_j z + \bar{\alpha}_j) p_{\mathsf{Hid}_j}(\tilde{\tau} z + \bar{\tau})(\tilde{\tau} z + \bar{\tau} - \mathsf{aux}_j)},
$$

$$s_{1,j} \leftarrow g_1^{(\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}})+(\tilde{\alpha}_j z+\bar{\alpha}_j)(\tilde{\nu}z+\bar{\nu})((\tilde{a}_j+\tilde{\rho}_j)z+\bar{a}_j+\bar{\rho}_j)p_{M_j}(\tilde{\tau}z+\bar{\tau})}, \quad s_{2,j} \leftarrow g_1^{(\tilde{a}_j+\tilde{\rho}_j)z+\bar{a}_j+\bar{\rho}_j}.$$

Note that the reduction avoids embedding an inverse challenge term in $s_{2,j}$, since the inverses of z in the exponent are not known. Still, the terms yield a valid and uniformly distributed response to the query, as

$$\mathbf{e}(g_1^{(\tilde{\tau}z+\bar{\tau}-\mathsf{aux}_j)}, C_j) = \mathbf{e}\left(g_1^{(\tilde{\tau}z+\bar{\tau}-\mathsf{aux}_j)(\tilde{\alpha}_j z+\bar{\alpha}_j)p_{\mathsf{Hid}_j}(\tilde{\tau}z+\bar{\tau})}, g_2^{(\tilde{\nu}z+\bar{\nu})p_{\mathsf{Rev}_j}(\tilde{\tau}z+\bar{\tau})}\right) = \mathbf{e}\left(\hat{C}_j, h_2^{p_{\mathsf{Rev}_j}(\tilde{\tau}z+\bar{\tau})}\right),$$

and

$$\begin{aligned}\mathbf{e}(s_{1,j}, g_2) =& \mathbf{e}\left(g_1^{(\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}})+(\tilde{\alpha}_j z+\bar{\alpha}_j)(\tilde{\nu}z+\bar{\nu})((\tilde{a}_j+\tilde{\rho}_j)z+\bar{a}_j+\bar{\rho}_j)p_{M_j}(\tilde{\tau}z+\bar{\tau})}, g_2\right) \\ =& g_T^{\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}}} \cdot \mathbf{e}\left(g_1^{(\tilde{a}_j+\tilde{\rho}_j)z+\bar{a}_j+\bar{\rho}_j}, g_2^{(\tilde{\nu}z+\bar{\nu})(\tilde{\alpha}_j z+\bar{\alpha}_j)p_{M_j}(\tilde{\tau}z+\bar{\tau})}\right) = \mathsf{vk} \cdot \mathbf{e}\left(s_{2,j}, C_j\right).\end{aligned}$$

Note that all of the above exponents are polynomials of degree at most $l+3$ in z, so $\mathcal{B}$ can compute them from its q-DL challenge. $\mathcal{B}$ embeds the q-DL challenge in f_P as described in Lemma 3.1. This yields a univariate polynomial f' in z. The following holds due to Lemma 3.1:

1. Since f_P has degree larger than 0, f' must be of the same degree.
2. The coefficient of f''s highest degree monomial is a polynomial in the $\tilde{x}$ values, that is, in $\tilde{\tau}, \tilde{\nu}, \tilde{\mathsf{sk}}$, $\{\tilde{\mathsf{sk}}_i\}_{i\in[t-1]}$, $\{\tilde{r}_i\}_{i\in[q_I]}$ and $\{\tilde{a}_j, \tilde{\alpha}_j, \tilde{\rho}_j\}_{j\in[q_P]}$.
3. The $\tilde{x}$ blinding terms hide z from $\mathcal{A}$'s view for each variable x, while $\bar{x}$ perfectly hides $\tilde{x}$.

Therefore, we can upper-bound the probability that f' is the zero polynomial even if f_P is not, using the Schwartz-Zippel Lemma. Since f' is a polynomial of degree at most $l+3+1+l = 2l+4$, it is non-zero with probability at least $1 - \frac{2l+4}{p}$, where p is the size of field $\mathbb{F}_p$. Using standard techniques [19], $\mathcal{B}$ can then factor f' for the challenge z.

Case 2: f_P is the zero polynomial If $f_P(\cdot) = 0$ for any choice of variables, all terms that contain a unique product of variables must have a coefficient of 0, as they cannot be cancelled by a sum of other terms in f_P. We enumerate the terms in the order in which they cancel and list each unique product of variables below.

$$\begin{aligned}f_P(\mathsf{sk}, \nu, \tau, \{r_i\}_{i\in[q_I]}, \{\alpha_j, \beta_j\}_{j\in[q_P]}) = &\Bigg(\cancel{u_0}^{(7)} + u_1\nu \sum_{i\in[l]} + \cancel{\mathbf{u}_2[i]\tau^i}^{(8)} + \sum_{i\in[l]} \mathbf{u}_3[i]\nu\tau^i \\ &+ \sum_{j\in[q_P]} \mathbf{u}_4[j]\nu\alpha_j p_{M_j}(\tau)\Bigg)(\tau - \mathsf{aux}) - \Bigg(u_0' + \cancel{u_1'\nu}^{(4)} + \sum_{i\in[l+1]} \mathbf{u}_2'[i]\tau^i + \sum_{i\in[l]} \cancel{\mathbf{u}_3'[i]\nu\tau^i}^{(4)} \\ &+ \sum_{i\in[q_I]} \cancel{\mathbf{u}_4'[i](\mathsf{sk} + \nu r_i p_i(\tau))}^{(5)} + \cancel{\mathbf{u}_5'[i]r_i}^{(3)} + \sum_{j\in[q_P]} \cancel{\mathbf{u}_6'[j](\mathsf{sk} + \nu\alpha_j\beta_j p_{M_j}(\tau))}^{(1)} + \cancel{\mathbf{u}_7'[j]\beta_j}^{(2)} \\ &+ \mathbf{u}_8'[j]\alpha_j p_{M_{\mathsf{Hid},j}}(\tau)(\tau - \mathsf{aux}_j) + \cancel{\mathbf{u}_9'[c]\mathsf{sk}_c}^{(6)}\Bigg)\nu p_{\mathsf{Rev}}(\tau).\end{aligned}$$

The following terms cancel in this equation:

#	Unique Variable Product	#	Unique Variable Product
(1)	$\nu^2\alpha_j\beta_j$	(5)	$\nu^2 r_i$
(2)	$\nu\beta_j$	(6)	$\nu\mathsf{sk}_c, c \in \mathcal{C}$
(3)	νr_i	(7)	1 (constant)
(4)	$\nu^2\tau^x, x \in [0, l]$	(8)	$\tau^x, x \in [0, l+1]$

The remaining polynomial is

$$f_P(\cdot) = \left(u_1\nu + \sum_{i\in[l]} \mathbf{u}_2[i]\nu\tau^i + \sum_{j\in[q_P]} \mathbf{u}_3[j]\nu\alpha_j p_{M_j}(\tau)\right)(\tau - \mathsf{aux})$$

$$- \left(u'_0 + \sum_{i \in [l]} \mathbf{u}'_2[i]\tau^i + \mathbf{u}'_8[j]\alpha_j p_{M_{\mathsf{Hid},j}}(\tau)(\tau - \mathsf{aux}_j) \right) \nu p_{\mathsf{Rev}}(\tau).$$

Combinations involving $\mathbf{u}'_8$ and $\mathbf{u}_3$ intuitively correspond to outputs that reuse commitments that were part of previous $\mathcal{O}^{\mathsf{tPresent}}$ queries. For all $j \in [q_P]$, for the α_j terms to cancel, it must hold that

$$\mathbf{u}_3[j]\nu p_{M_j}(\tau)(\tau - \mathsf{aux}) = \mathbf{u}'_8[j]\nu p_{\mathsf{Hid}_j}(\tau) p_{\mathsf{Rev}}(\tau)(\tau - \mathsf{aux}_j).$$

By the abort condition of $\mathcal{O}^{\mathsf{tPresent}}$, it must hold that $M_j \cap \mathcal{X}_{\mathsf{aux}} = \emptyset$ and $\mathsf{aux}_j \in \mathcal{X}_{\mathsf{aux}}$. Therefore, the τ terms on both sides of the equation can only cancel, if $\mathsf{aux} = \mathsf{aux}_j$ and $p_{M_{\mathsf{Rev}}}(\tau) = p_{M_{\mathsf{Rev},j}}(\tau)$ or vice versa. This is ruled out by assumption, implying that $\mathbf{u}_3[j] = \mathbf{u}'_8[j] = 0$ for all $j \in [q_P]$. Therefore, C can be represented only using coefficients u_1 and $\mathbf{u}_2$.

Ext outputs $p(\tau) = u_1 + \sum_{i \in [l]} \mathbf{u}_2[i]\tau^i$, for which $C = h_2^{p(\tau)}$. From the above equation, it holds that $p(\tau)(\tau - \mathsf{aux}) = \left(u'_0 + \sum_{i \in [l]} \mathbf{u}'_2[i]\tau^i\right) p_{\mathsf{Rev}}(\tau)$. As $\mathsf{aux} \in \mathcal{X}_{\mathsf{aux}}$ cannot be a root of p_{Rev}, $p_{\mathsf{Rev}} | p$. Only in case 1 is Ext not successful. In this case, we can break the $\mathsf{q\text{-}DL}$ assumption with probability $1 - \left(\mathsf{Adv}^{\mathsf{q\text{-}DL}}_{\mathcal{B}}(\lambda) + \frac{2l+4}{p}\right)$. Therefore, if the $\mathsf{q\text{-}DL}$ assumption holds, Ext has a success probability of $1 - \mathsf{negl}(\lambda)$. □

Remark B.1 (Independence of forgeries following Lemma 7.2)**.** *The extractor in Lemma 7.2 outputs a polynomial p such that $C = h_2^{p(\tau)}$. Importantly, p is a polynomial only in τ and does not depend on the secret values used by the oracles to which* Ext *has access. This is inferred from the proof, as all terms that involve previous oracle queries cancel.*

Theorem 7.4 (restated). *If the bilinear* $\mathsf{q\text{-}DL}$ *assumption holds, our credential construction* BBCred *fulfills unforgeability as per Definition 4.3 in the algebraic group model.*

We show the theorem using the extractability of presentations shown in Lemma 7.2. We then show that the forgery output by the adversary can be used to break the $\mathsf{q\text{-}DL}$ assumption in the AGM. In detail, we prove that for all PPT adversaries $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$, such that for all $\lambda, n, l \in \mathbb{N}$, it holds that

$$\mathsf{Adv}^{\mathsf{tAC\text{-}UF}}_{\mathcal{A},\mathrm{BBC}_{\mathrm{RED}}}(\lambda, n, l) \leq \mathsf{Adv}^{\mathsf{q\text{-}DL}}_{\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda),$$

where $\mathsf{tAC\text{-}UF}$ is defined in Definition 4.3. We show the statement via a game-hop, before giving a final reduction to $\mathsf{q\text{-}DL}$.

Game_1: Game_1 is equivalent to Game_0, except for the following change: instead of outputting a presentation forgery of the form

$$(\sigma_1, \sigma_2, C, \hat{C}, \mathsf{aux}, \mathsf{Rev}),$$

the adversary outputs

$$(\sigma_1, \sigma_2, p, \mathsf{aux}, \mathsf{Rev}),$$

where p is the polynomial encoded in C. We build a PPT adversary $\mathcal{B}$ in Game_1, whose success probability differs from that of $\mathcal{A}$ only by a negligible amount. Let $\mathcal{B}$ simulate Game_0 to $\mathcal{A}$ by relaying all setup parameters, oracle queries, and responses from its own challenger to $\mathcal{A}$. When $\mathcal{A}$ outputs a presentation of the form $(\sigma_1, \sigma_2, C, \hat{C}, \mathsf{aux}, \mathsf{Rev})$, it must hold that

$$e(g_1^{\tau - \mathsf{aux}}, C) = e(\hat{C}, h_2^{p_{\mathsf{Rev}}(\tau)}).$$

$\mathcal{B}$ can use the extractor given in Lemma 7.2 to extract a polynomial p from $(C, \hat{C}, \mathsf{aux}, \mathsf{Rev})$ such that $C = h_2^{p(\tau)}$ and $p_{\mathsf{Rev}} \mid p$ with overwhelming probability. It follows from Lemma 7.2 that

$$\mathsf{Adv}^{\mathrm{Game}_0}_{\mathcal{A},\mathrm{BBC}_{\mathrm{RED}}}(\lambda, n, l) \leq \mathsf{Adv}^{\mathrm{Game}_1}_{\mathcal{B},\mathrm{BBC}_{\mathrm{RED}}}(\lambda, n, l) + \mathsf{negl}(\lambda).$$

GAME$_2$: GAME$_2$ is equivalent to GAME$_1$, except for the following change: the algebraic adversary wins only if the polynomial induced by the forged signature verification equation is the zero polynomial. Let $\mathcal{B}$ make q_I queries to the partial issuance oracle and q_P queries to the threshold presentation oracle. In the AGM, $\mathcal{B}$ additionally outputs coefficient vectors $\mathbf{u}$ and $\mathbf{u}'$ describing σ_1 and σ_2 as linear combinations of previously seen elements in $\mathbb{G}_1$. As in the proof of Lemma 7.2, for the j-th presentation query define $\beta_j := \frac{\rho_j + a_j}{\alpha_j}$. We may regard β_j as uniformly random and independent of α_j. The set of group elements available to $\mathcal{B}$ is therefore

$$S = \Big\{ g_1, h_1, \{g_1^{\tau^i}\}_{i\in[l+1]}, \{h_1^{\tau^i}\}_{i\in[l]}, \{g_1^{\mathsf{sk}_{I(i)}} h_1^{r_i p_i(\tau)}, g_1^{r_i}\}_{i\in[q_I]}, \{g_1^{\mathsf{sk}_c}\}_{c\in\mathcal{C}}, \\ \{g_1^{\mathsf{sk}} h_1^{\alpha_j \beta_j p_{M_j}(\tau)}, g_1^{\beta_j}, g_1^{\alpha_j p_{\mathsf{Hid},j}(\tau)(\tau - \mathsf{aux}_j)}\}_{j\in[q_P]} \Big\},$$

where $\mathcal{I}(x) : [q_I] \to [n]$ maps a queried credential to its signer. For $\mathcal{B}$ to win the game, it must hold that $\mathbf{e}(\sigma_1, g_2) = \mathsf{vk} \cdot \mathbf{e}(\sigma_2, h_2^{p(\tau)})$, where $h_2^{p(\tau)} = C$ by Lemma 7.2. The pairing check implies the following polynomial equation in the exponent:

$$\begin{aligned} f(\{\mathsf{sk}_i\}_{i\in[n]}, \nu, \tau, \{r_i\}_{i\in[q_I]}, \{\alpha_j, \beta_j\}_{j\in[q_P]}) = & -\mathsf{sk} + u_0 + u_1\nu + \sum_{i\in[l+1]} u_2[i]\tau^i + \sum_{i\in[l]} u_3[i]\nu\tau^i \\ & + \sum_{i\in[q_I]} u_4[i]\big(\mathsf{sk}_{I(i)} + \nu r_i p_i(\tau)\big) + \sum_{i\in[q_I]} u_5[i] r_i + \sum_{c\in\mathcal{C}} u_6[c]\mathsf{sk}_c + \sum_{j\in[q_P]} u_7[j]\big(\mathsf{sk} + \nu\alpha_j\beta_j p_{M_j}(\tau)\big) \\ & + \sum_{j\in[q_P]} u_8[j]\beta_j + \sum_{j\in[q_P]} u_9[j]\alpha_j p_{\mathsf{Hid},j}(\tau)(\tau - \mathsf{aux}_j) - \nu p(\tau)\Bigg(u_0' + u_1'\nu + \sum_{i\in[l+1]} u_2'[i]\tau^i + \sum_{i\in[l]} u_3'[i]\nu\tau^i \\ & + \sum_{i\in[q_I]} u_4'[i]\big(\mathsf{sk}_{I(i)} + \nu r_i p_i(\tau)\big) + \sum_{i\in[q_I]} u_5'[i] r_i + \sum_{c\in\mathcal{C}} u_6'[c]\mathsf{sk}_c + \sum_{j\in[q_P]} u_7'[j]\big(\mathsf{sk} + \nu\alpha_j\beta_j p_{M_j}(\tau)\big) \\ & + \sum_{j\in[q_P]} u_8'[j]\beta_j + \sum_{j\in[q_P]} u_9'[j]\alpha_j p_{\mathsf{Hid},j}(\tau)(\tau - \mathsf{aux}_j)\Bigg). \end{aligned}$$

$\mathcal{B}$ is successful in Game$_1$ iff f evaluates to zero for the specific choice of exponents defined in the game.

We next bound the probability that $\mathcal{B}$ wins GAME$_1$ but not GAME$_2$, i.e., that it outputs a valid forgery for which f is non-zero. Assume that this event occurs. Then we can build a reduction $\mathcal{D}$ for q-DL that receives as input a challenge vector $(\mathsf{BG}, \{g_1^{z^i}, g_2^{z^i}\}_{i\in[l+3]})$ (i.e. $q = l + 3$). As in the proof of Lemma 7.1 and Lemma 7.2, $\mathcal{D}$ embeds the challenge z into every variable x of f by setting $x := \tilde{x}z + \bar{x}$, where the pair $\tilde{x}, \bar{x} \xleftarrow{\$} \mathbb{F}_p$ is sampled independently for each variable. This blinding technique follows the approach of [34]. $\mathcal{D}$ additionally samples $\tilde{\tau}, \bar{\tau}, \tilde{\nu}, \bar{\nu} \xleftarrow{\$} \mathbb{F}_p$ and outputs the BBCRED public parameters as

$$\mathsf{pp} = \Big(\mathsf{BG}, g_1^{\tilde{\nu}z+\bar{\nu}}, g_2^{\tilde{\nu}z+\bar{\nu}}, \{g_1^{(\tilde{\tau}z+\bar{\tau})^i}, g_2^{(\tilde{\tau}z+\bar{\tau})^i}\}_{i\in[l+1]}, \{g_1^{(\tilde{\nu}z+\bar{\nu})\cdot(\tilde{\tau}z+\bar{\tau})^i}, g_2^{(\tilde{\nu}z+\bar{\nu})\cdot(\tilde{\tau}z+\bar{\tau})^i}\}_{i\in[l]}\Big).$$

By doing so, $\mathcal{D}$ embeds the randomized challenge in the exponent of the elements in the CRS. For key generation, $\mathcal{D}$ samples $\tilde{\mathsf{sk}}_i, \bar{\mathsf{sk}}_i \xleftarrow{\$} \mathbb{F}_p$, $\forall i \in [n]$, and $\tilde{\mathsf{sk}}, \bar{\mathsf{sk}} \xleftarrow{\$} \mathbb{F}_p$. It then simulates Shamir secret sharing by choosing a function $F(Z) := (\tilde{\mathsf{sk}}z + \bar{\mathsf{sk}}) + \sum_{i=1}^{t-1}(\tilde{\mathsf{sk}}_i z + \bar{\mathsf{sk}}_i)Z^i$, chooses $\mathsf{sk}_i = F(i)$, and sets $g_1^{\mathsf{sk}_i}$ as the secret key of party i. The simulated shares have the same structure as regular Shamir secret shares. Since F is a polynomial of degree $t-1$, up to $t-1$ shares do not allow an adversary to learn anything about the secret. Also, for $I \subseteq [n]$, $|I| \geq t$, it holds that $(\tilde{\mathsf{sk}}z + \bar{\mathsf{sk}}) = \sum_{i\in I} \lambda_i F(i)$. $\mathcal{D}$ then sets $\mathsf{vk} \leftarrow \mathbf{e}(g_1^{\tilde{\mathsf{sk}}z+\bar{\mathsf{sk}}}, g_2)$.

Upon receiving the ith issuance query on a polynomial p_i, $\mathcal{D}$ samples $\tilde{r}_i, \bar{r}_i \xleftarrow{\$} \mathbb{F}_p$ and uses $\tilde{r}_i z + \bar{r}_i$, a masked q-DL challenge value, for signature generation. $\mathcal{D}$ can compute a valid signature given the q-DL challenge as follows:

$$\sigma_{1,i} \leftarrow g_1^{\mathsf{sk}_{\mathcal{I}(i)}} \cdot g_1^{p_i(\tilde{\tau}z+\bar{\tau})\cdot(\tilde{r}_i z+\bar{r}_i)(\tilde{\nu}z+\bar{\nu})}, \qquad \sigma_{2,i} \leftarrow g_1^{\tilde{r}_i z+\bar{r}_i}.$$

$\mathcal{D}$ can easily answer corruption queries by revealing $g_1^{\mathsf{sk}_i}$. Furthermore, $\mathcal{D}$ can simulate presentation queries as in Lemma 7.1. In particular, $\mathcal{D}$ samples affine masks for α_j and β_j and computes

$$C_j = g_2^{(\tilde{\nu}z+\bar{\nu})(\tilde{\alpha}_j z+\bar{\alpha}_j)p_{M_j}(\tilde{\tau}z+\bar{\tau})},$$

$$\widehat{C}_j = g_1^{(\tilde{\alpha}_j z+\bar{\alpha}_j)p_{\mathsf{Hid},j}(\tilde{\tau}z+\bar{\tau})(\tilde{\tau}z+\bar{\tau}-\mathsf{aux}_j)},$$

$$s_{1,j} = g_1^{(\widetilde{\mathsf{sk}}z+\overline{\mathsf{sk}})+(\widetilde{\nu}z+\overline{\nu})(\widetilde{\alpha}_j z+\overline{\alpha}_j)(\widetilde{\beta}_j z+\overline{\beta}_j)p_{M_j}(\widetilde{\tau}z+\overline{\tau})},$$

and

$$s_{2,j} = g_1^{\widetilde{\beta}_j z+\overline{\beta}_j}.$$

As in Lemma 7.1, these values form a valid and uniformly distributed presentation response. Therefore, $\mathcal{D}$ perfectly simulates the view of $\mathcal{B}$ in GAME_1.

We next show that whenever $\mathcal{B}$ outputs a valid forgery for which f is non-zero, $\mathcal{D}$ can solve its q-DL challenge. Recall that, for every variable x occurring in f, $\mathcal{D}$ samples independent $\widetilde{x}, \overline{x} \xleftarrow{\$} \mathbb{F}_p$ and embeds $x = \widetilde{x}z + \overline{x}$. Let Z be a formal indeterminate and define the univariate polynomial

$$f'(Z) := f\big(\widetilde{x}_1 Z + \overline{x}_1, \ldots, \widetilde{x}_m Z + \overline{x}_m\big),$$

where $x_1, \ldots, x_m$ are all variables occurring in f. Thus, evaluating f' at the q-DL challenge z gives exactly the polynomial relation induced by the simulated execution. Since the forged signature verifies, the induced exponent relation vanishes, and hence $f'(z) = 0$. Moreover, by construction, $\deg(f') \le 2l+4$. Since f is non-zero, Lemma 3.1 implies that the leading coefficient of f' is a non-zero polynomial in the blinding values

$$\widetilde{\tau}, \widetilde{\nu}, \widetilde{\mathsf{sk}}, \{\widetilde{s}_k\}_{k\in[t-1]}, \{\widetilde{r}_i\}_{i\in[q_I]}, \{\widetilde{\alpha}_j, \widetilde{\beta}_j\}_{j\in[q_P]},$$

of degree at most $2l + 4$. These blinding values are uniformly random and perfectly hidden from the view of $\mathcal{B}$. Therefore, by the Schwartz–Zippel lemma, the leading coefficient vanishes with probability at most $\frac{2l+4}{p}$. Hence, except with probability at most $(2l + 4)/p$, the polynomial f' is non-zero. Thus, since $f'(z) = 0$ and f' is of degree at most $2\ell + 4$, $\mathcal{D}$ can compute the roots of f' in polynomial time and identify the challenge z among them. Therefore, it follows that

$$\mathsf{Adv}^{\text{GAME}_1}_{\mathcal{B},\text{BBCRED}}(\lambda, n, l) \le \mathsf{Adv}^{\text{GAME}_2}_{\mathcal{B},\text{BBCRED}}(\lambda, n, l) + \mathsf{Adv}^{\mathsf{q\text{-}DL}}_{\mathcal{D}}(\lambda) + \mathsf{negl}(\lambda).$$

It remains to show that

$$\mathsf{Adv}^{\text{GAME}_2}_{\mathcal{B},\text{BBCRED}}(\lambda, n, l) = 0.$$

In order to show this, we prove that f cannot be the zero polynomial. We use the same technique as before, where we eliminate unique monomials. We first eliminate all terms originating from previous presentation queries. Fix $j \in [q_P]$. The only terms containing the product $\nu^2\alpha_j\beta_j$ arise from $-\nu p(\tau)u'_7[j]\nu\alpha_j\beta_j p_{M_j}(\tau)$, and therefore, if $p \neq 0$, $u'_7[j] = 0$. The remaining $\nu\alpha_j\beta_j$ term then implies $u_7[j] = 0$. Likewise, the remaining term containing β_j is $u_8[j]\beta_j - u'_8[j]\nu p(\tau)\beta_j$, which implies $u_8[j] = u'_8[j] = 0$ when $p \neq 0$. Finally, the terms containing α_j is

$$u_9[j]\alpha_j p_{\mathsf{Hid},j}(\tau)(\tau - \mathsf{aux}_j) - u'_9[j]\nu p(\tau)\alpha_j p_{\mathsf{Hid},j}(\tau)(\tau - \mathsf{aux}_j),$$

and hence $u_9[j] = u'_9[j] = 0$.

Thus, for every $j \in [q_P]$, all terms originating from previous $O^{t\mathsf{Present}}$ queries make no contribution to f.

Consequently, the remaining polynomial is

$$\begin{aligned} f(\cdot) = &-\mathsf{sk} + u_0 + u_1\nu + \sum_{i\in[l+1]} u_2[i]\tau^i + \sum_{i\in[l]} u_3[i]\nu\tau^i + \sum_{i\in[q_I]} \Big(u_4[i](\mathsf{sk}_{I(i)} + \nu r_i p_i(\tau)) + u_5[i]r_i\Big) \\ &+ \sum_{c\in\mathcal{C}} u_6[c]\mathsf{sk}_c - \nu p(\tau)\Bigg(u'_0 + u'_1\nu + \sum_{i\in[l+1]} u'_2[i]\tau^i + \sum_{i\in[l]} u'_3[i]\nu\tau^i \\ &+ \sum_{i\in[q_I]} \Big(u'_4[i](\mathsf{sk}_{I(i)} + \nu r_i p_i(\tau)) + u'_5[i]r_i\Big) + \sum_{c\in\mathcal{C}} u'_6[c]\mathsf{sk}_c\Bigg). \end{aligned}$$

Similarly, all unique products of variables must have coefficient zero. The terms containing $\nu^2 r_i$ imply $u'_4[i] = 0$ and the terms containing $\mathsf{sk}_c\nu$ imply $u'_6[c] = 0$. To cancel the terms containing νr_i, it must hold for all $i \in [q_I]$ that

$$u_4[i]p_i(\tau) - u'_5[i]p(\tau) = 0.$$

If $u_4[i] = u'_5[i] = 0$, the i-th issuance query does not contribute. Otherwise, since $u_4[i]$ and $u'_5[i]$ are field elements, the above polynomial identity implies $p_i = c_i p$ for some $c_i \in \mathbb{F}_p^*$. In particular, p_i and p have the same roots. Since Lemma 7.2 gives $p_{\mathsf{Rev}^*} \mid p$, it follows that $\mathsf{Rev}^* \subseteq M_i$. Moreover, all issuance queries

with non-zero $u_4[i]$ have the same root set, and therefore correspond to the same attribute set; denote it by M^*. By the winning condition of the tAC-UF game, $|\mathcal{C} \cup \mathcal{U}_{\mathsf{iss}}[M^*]| < t$. Hence the signing-key shares that occur with non-zero coefficients in f belong to fewer than t distinct Shamir shares.

By the privacy of Shamir secret sharing, any set of fewer than t shares is independent of the master secret sk. Consequently,

$$\sum_{c \in C} u_6[c]\mathsf{sk}_c + \sum_{i \in [q_I]} u_4[i]\mathsf{sk}_{I(i)}$$

cannot identically cancel the remaining $-\mathsf{sk}$ term. This contradicts the assumption that f is the zero polynomial. Therefore,

$$\mathsf{Adv}^{\mathrm{GAME}_2}_{\mathcal{B},\mathrm{BBC}_{\mathrm{RED}}}(\lambda, n, l) = 0.$$

Combining the game hops yields

$$\mathsf{Adv}^{\mathrm{GAME}_0}_{\mathcal{A},\mathrm{BBC}_{\mathrm{RED}}}(\lambda, n, l) \leq \mathsf{Adv}^{\mathrm{GAME}_2}_{\mathcal{B},\mathrm{BBC}_{\mathrm{RED}}}(\lambda, n, l) + \mathsf{Adv}^{\mathsf{q\text{-}DL}}_{\mathcal{D}}(\lambda) + \mathsf{negl}(\lambda) \leq \mathsf{negl}(\lambda),$$

which completes the proof.