CAnonize: A Compact Anonymous Survey Protocol

Marie Lonfils
UCLouvain
marie.lonfils@uclouvain.be

Olivier Pereira
UCLouvain
olivier.pereira@uclouvain.be

Thomas Peters
UCLouvain
thomas.peters@uclouvain.be

Moti Yung
Google LLC - Columbia University
moti.yung@google.com

Abstract

Online surveys are frequently used to collect feedback on sensitive topics, yet most deployed platforms rely on authenticated accounts, cookies, or operator-enforced policies to protect anonymity and limit participation. These approaches offer limited protection against retrospective deanonymization in the event of database compromise. The anonymous survey primitive addresses this problem by enabling authenticated yet anonymous submissions with at-most-once participation per survey, without requiring a trusted setup. Current solutions may however be computationally demanding when aiming for large scale deployment.

We revisit this primitive and present CAnonize, a new anonymous survey protocol that considerably improves the efficiency compared to the state of the art, while also relying on weaker computational assumptions and preserving a strong corruption model: our protocol relies solely on the SXDH assumption in asymmetric bilinear groups (q-type assumptions were used before), and offers unlinkability under adaptive corruption of the registration authority, survey authorities, and users, even under full transcript exposure and post-compromise database leakage.

We confirm the efficiency benefits and practicality of CAnonize through a prototype implementation in Rust: using the BLS12-381 curves, the running times for submitting and processing a survey response are below 100ms for the user and for the survey authority.

Keywords

anonymous surveys, privacy enhancing technologies, pairing-based cryptography, Groth–Sahai proofs, structure-preserving signatures

1 Introduction

Online surveys are widely used to collect feedback on sensitive topics, including political opinions, health conditions, workplace misconduct, and regulatory compliance. In such settings, response accuracy depends critically on respondents' perceived anonymity. Empirical evidence demonstrates that stronger anonymity and confidentiality guarantees increase truthful disclosure, in some cases by double-digit percentage points compared to identifiable settings [38, 47].

In practice, however, most survey platforms rely on access control, authenticated accounts, or operator-enforced policies to limit participation and protect confidentiality. Even when responses are stored without explicit identifiers, operators frequently retain metadata (e.g., login state, IP addresses, device identifiers) that can enable retrospective linkage. Moreover, large-scale data breaches routinely expose stored personal information due to external compromise, misconfiguration, or insider misuse [54], while regulatory frameworks increasingly constrain the collection and retention of personally identifiable information [36, 57].

These considerations motivate cryptographic survey mechanisms that provide anonymity and participation integrity *by design*, such that compromise of stored survey data does not enable retrospective deanonymization and correctness does not rely on a trusted online authority.

Anonymous Surveys. Hohenberger *et al.* formalized the primitive of authenticated yet anonymous surveys without trusted third parties and, as a first instantiation, proposed the *Anonize* anonymous survey protocol [35]. Anonize achieves authenticated yet anonymous survey participation without trusted setup and was deployed in the Brave browser as part of the Brave Rewards system to compensate content creators for advertising [1], as well as in universities for class surveys. Anonize, however, remains fairly demanding from a computational point of view, and Brave moved to other solutions (Privacy-Pass for user authentication combined with STAR for telemetry), citing the computational requirements as one factor motivating the adoption of a different approach (the need for different features more adapted to telemetry rather than surveys was another) [25].

An anonymous survey protocol involves three roles: a registration authority (RA), users, and survey authorities (SA). Users obtain credentials from the RA. Any user may act as an SA and instantiate a survey targeting a dynamically defined subset of registered users. Eligible users can then submit exactly one response in a non-interactive and anonymous manner.

An anonymous survey scheme aims to provide three core properties:

- *Unlinkability.* For any two honest users eligible for a survey, their responses are computationally indistinguishable, even if the RA, the SAs, and any subset of users are adaptively corrupted and colluding.
- *Correctness.* Only authorized users can submit responses to a given survey.
- *Security against malicious users.* Each authorized user can contribute at most one response per survey.

Proceedings on Privacy Enhancing Technologies 2026(4), 396–414

https://doi.org/10.56553/popets-2026-0127

We use the strong corruption model of Hohenberger et al. [35] used for Anonize, in which the RA, SAs, and users may be adaptively corrupted. In particular, unlinkability must hold even if survey transcripts and stored data are later exposed. That is, compromise of the survey database after execution must not enable retrospective deanonymization of honest participants.

These properties ensure that participants' identities remain protected while preventing unauthorized or duplicate submissions. While these guarantees hold at the cryptographic level, the protocol does not hide network-level metadata such as IP addresses or timing information. Mitigating these risks requires complementary network-level anonymity mechanisms (e.g., mixnets or Tor), which are assumed to be available separately. Anonize demonstrated the feasibility of this primitive and motivated real-world deployments. However, its computational cost and proof sizes are fairly large, coming from the combination of blind signatures, the Dodis–Yampolskiy PRF, and zero-knowledge proofs requiring straight-line extraction.

This motivates the following question:

Can substantially more efficient authenticated anonymous surveys with single participation be realized?

As a side question, we are also interested in achieving this goal while also offering stronger security guarantees. Anonize, which combines several distinct primitives, relies on multiple computational assumptions, including ad-hoc assumptions, in the random oracle model (ROM). We would be interested in designing a mechanism that would rely on a single standard computational assumption (still in the ROM).

Our Contributions. In this work, we design CAnonize, with the goal of offering considerably better efficiency compared to the state-of-the-art solution, Anonize. At the same time, we also offer the same privacy guarantees under weaker computational assumptions.

CAnonize redesigns the credential issuance and the survey submission mechanisms. It is pairing-based and relies solely on the SXDH assumption in asymmetric bilinear groups. We work in the common reference string (CRS) model for Groth–Sahai proofs and analyze hash-based components in the random oracle model. CAnonize relies on a CRS, but it is public-coin and can even be derived from the random oracle. The CRS does not require any trusted setup and cannot embed any trapdoor. Furthermore, we only make a mild use of the random oracle model that does not require rewinding.

More concretely, our contributions are:

- **Novel scheme structure.** We introduce a fresh CRS derivation process applied for each new response submission to ensure secure concurrent extraction in the security model of Anonize.
- **Efficiency improvements.** For 128-bit security instantiated over BLS12-381, CAnonize theoretically reduces total communication per survey instance by 92% and end-to-end execution time by 70% compared to Anonize.
- **Weaker computational assumptions.** CAnonize only relies on the SXDH assumption in asymmetric bilinear groups, in contrast to Anonize which relied on various assumptions, including q-type assumptions.
- **Security analysis.** We prove that CAnonize satisfies the three core security properties—unlinkability, correctness, and security against malicious users—under SXDH in the CRS model and ROM.
- **Prototype implementation.** We implement the complete CAnonize workflow for a single survey instance in Rust using `arkworks` over the BLS12-381 curve. Our prototype includes credential issuance, survey creation, authorization, submission, and verification, and empirically confirms the predicted performance gains relative to a re-implementation of Anonize with equivalent security parameters. For instance, the survey response submission of CAnonize requires $\approx$ 10% of the bandwidth of our optimized re-implementation of Anonize, and $\approx$ 24% and $\approx$ 38% of the user and server computational requirements.

One of the bottlenecks of Anonize (in bandwidth and time) comes from the requirement of straight-line extraction. In the Anonize paper, this is achieved using a transform due to Pass [48], but the Brave implementation used the more efficient transform due to Fischlin instead [28]. Since then, the Fischlin transform has remained an active research area, and to make our comparison more robust, our Anonize re-implementation is based on the state-of-the-art choice of parameters for the Fischlin transform proposed by Chen and Lindell [21]. As a result, our baseline is already more efficient than previously existing parameterization for a given security level.

Technical Overview. At a high level, CAnonize re-architects three components of Anonize. First, we employ Groth–Sahai proofs to obtain efficient non-interactive proofs for pairing-product and multi-scalar multiplication equations in bilinear groups. Second, we replace blind signatures with a structure-preserving signature scheme that integrates naturally with pairing-based proof systems, enabling efficient proof of credential validity within the Groth–Sahai framework. Third, we substitute the Dodis–Yampolskiy PRF with a Naor–Pinkas–Reingold PRF, reducing computational overhead while maintaining pseudorandomness under SXDH.

A key technical challenge is binding credential presentations to survey identifiers in a way that enforces single participation without introducing cross-survey linkability. Single participation is enforced via a survey-specific pseudorandom tag derived from a Naor–Pinkas–Reingold PRF, proven correct and well-formed within the Groth–Sahai framework. Additionally, we carefully manage the use of the Groth–Sahai CRS to ensure sound extraction while preserving unlinkability, without introducing trusted online trapdoor holders, by introducing a per-submission CRS derivation mechanism allowing concurrent extraction.

A significant source of efficiency benefits of CAnonize is also the removal of the need for proofs offering straight-line extraction, a key bottleneck of Anonize that was also previously identified by Kondi and Shelat [40]. In particular, the use of Groth–Sahai proofs with a structure-preserving signature removes the need for rewinding in the security proofs that was required by the blind signature scheme used in Anonize.

Organization. The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces cryptographic assumptions and building blocks. Section 4 formalizes the anonymous survey abstraction. Section 5 presents CAnonize and its security analysis. Section 6 evaluates performance. Section 7

concludes. Technical details on the Groth–Sahai proofs appear in Appendix A and Appendix B. The proof of security against malicious users is in Appendix C.

2 Related work

We position anonymous surveys relative to group, ring, and linkable signatures, anonymous credentials, blacklisting techniques, e-voting systems, anonymous data collection systems, and commercial survey platforms.

Group, Ring and Linkable Signatures. Group signatures [14, 20] provide signer anonymity within an explicitly specified set of public keys. Closer to our anonymous survey architecture and especially to its registration layer, Zhang *et al.* [60] introduced a group signature with certified limited opening based on a two-authority credential infrastructure, with one for group membership and a second for certifying a context-specific event, as in anonymous surveys. In their construction, the key generation center issues a credential binding the user identity to a public encoding of a user-generated secret value. Our registration authority similarly certifies a user identity together with an encoded secret PRF seed. The key distinction is that, in their construction, the second authority controls signature opening, whereas in our system it governs participation in the survey. Ring signatures [12, 50, 51] additionally enable an "ad-hoc" selection of the public keys set. List and linkable ring signatures [18, 29, 41, 42, 55] further support detection of repeated signatures by the same signer relative to a fixed tag. Our scheme leverages this idea of a per-context deterministic token that is unlinkable across contexts to detect duplicate participation. Constant-size linkable ring signatures [7, 8, 26, 58] are often constructed using accumulators, and membership is proven using a witness. Conceptually, the SA-issued per-user credential in our scheme plays a role analogous to a membership witness. However, adding a new member requires at least sublinear computational overhead in the number of users [49] or, to update a public key for each new member and the witnesses at a cost proportional to the number of new members since the last update [16]. This limitation makes traditional accumulator-based approaches less practical for dynamic surveys where authorized participants may be added at any time.

Anonymous Credentials and Blacklisting. Anonymous credential systems [10, 15] allow users to prove possession of certified attributes without revealing their identity. Modern constructions such as Idemix [17] provide selective disclosure and unlinkable repeated presentations of the same credential, properties that are closely aligned with the anonymous survey setting. However, standard anonymous credentials do not natively enforce *at-most-once participation per context.* Anonymous blacklisting and revocation schemes [34, 43, 52] enable users to prove that they are not on a blacklist while preserving anonymity, thereby supporting access control in privacy-preserving systems. Yet, blacklist-based approaches generally rely on a trusted authority to maintain revocation state or incur overhead proportional to the size of the blacklist. Moreover, they are not designed to enforce single participation per context without maintaining state that grows with the number of users or submissions, and may introduce cross-context linkability. In contrast, anonymous surveys require that credential presentations remain unlinkable across distinct surveys while guaranteeing exactly one valid submission per survey. CAnonize achieves this property with stateless verification, constant-size submissions, and without per-survey revocation lists or overhead that scales with the user population.

Electronic Voting Systems. Cryptographic e-voting systems [3, 22, 39] provide ballot privacy and election integrity, often with single-participation enforcement, public verifiability, and universal auditability. As in our construction, single-participation guarantees can be achieved using linkable tags. While e-voting and anonymous surveys share high-level goals, their threat models and system assumptions differ substantially. E-voting schemes typically assume a fixed voter roll, a global bulletin board, and dedicated tallying infrastructure (e.g., mixnets or homomorphic aggregation). They aim to provide public verifiability of election outcomes. In contrast, anonymous surveys target deployment scenarios with dynamically defined subsets of users and without universal verifiability, or complex tallying procedures.

Anonymous Data Collection Systems. Anonymous data collection systems such as PRIO [23], CONAN [61], and Privacy Pass and STAR [24] replaced Anonize in Brave's deployment and adapted related techniques to issue unlinkable tokens for bypassing repeated challenges such as CAPTCHAs. Unlike anonymous surveys, these anonymous data collection systems are designed to allow multiple token redemptions and therefore do not natively enforce at-most-once participation per context. This tradeoff better matches Brave's deployment requirements, which prioritize efficiency and telemetry functionality over strict single-participation guarantees.

A large body of work additionally studies private aggregation mechanisms for collecting numerical or structured data [30, 44], particularly in settings such as federated learning, IoT systems [4, 59], healthcare [5], and big data analytics [33]. These approaches commonly rely on techniques such as homomorphic encryption or differential privacy to compute aggregate statistics while protecting individual inputs. They are primarily designed for structured numerical aggregation tasks rather than anonymous collection of free-form textual responses as considered in survey systems.

Commercial Survey Platforms. Widely used survey platforms (e.g., Google Forms, SurveyMonkey, LimeSurvey, JotForm) enforce response limits via authenticated accounts, cookies, or IP-based mechanisms. These approaches inherently link responses to identifiers or metadata controlled by the platform operator. Even when identifying fields are omitted from survey responses, the operator typically retains sufficient auxiliary information to enable retrospective linkage in the event of compromise. Some platforms advertise encryption of stored responses, but do not provide cryptographic guarantees of anonymity against a malicious or compromised operator. In contrast, anonymous survey systems enforce anonymity and single participation at the cryptographic level, ensuring that compromise of stored survey data does not enable deanonymization of past responses.

3 Preliminaries

We introduce the notation (3.1) and the cryptographic primitives used in CAnonize, namely the SXDH assumption (3.2), structure-preserving signatures (3.3), Groth–Sahai proofs (3.4), the Naor–Pinkas–Reingold PRF (3.5), and a one-time signature scheme (3.6). We also describe the Fischlin transform (3.7), which we use in our re-implementation of Anonize to obtain non-interactive proofs of knowledge with straight-line extraction.

3.1 Notation

We consider:

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ as cyclic groups of prime order q for a λ-bit prime q.
- $g, \hat{g}$ as generators of respectively $\mathbb{G}_1$ and $\mathbb{G}_2$.
- $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ as an efficiently computable non-degenerate bilinear map.

Together, these elements form the scheme's public parameters $pp = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \hat{g})$, defining a prime order bilinear group satisfying the SXDH assumption.
We denote elements of $\mathbb{G}_2$ with a hat, e.g. $\hat{g}$.

3.2 SXDH Assumption

The proposed survey scheme relies on the SXDH assumption.

Definition 1. *The **Decisional Diffie-Hellman** (DDH) problem is hard in $\mathbb{G}$, a cyclic group of order q with generator g, if the distributions (g^a, g^b, g^{ab}) and (g^a, g^b, g^c), with $a, b, c \leftarrow \mathbb{Z}_q$, are indistinguishable by non-uniform polynomial-time adversaries.*[1]

Definition 2. *The **Symmetric external Diffie-Hellman** (SXDH) assumption assumes the hardness of DDH in both $\mathbb{G}_1$ and $\mathbb{G}_2$, in asymmetric pairing configurations.*

3.3 Structure-Preserving Signature

Structure-preserving signatures [2] are a class of digital signature schemes defined entirely in terms of group elements: the public keys, messages, and signatures are all elements of groups supporting efficient bilinear pairings, and the verification equations consist of pairing product equations. This structure allows for efficient composition with other cryptographic primitives that also operate on group elements without requiring additional encoding or hashing steps.

Jutla and Roy proposed a structure-preserving signature scheme [37] that integrates well with Groth–Sahai proofs. To the best of our knowledge, this signature is the most efficient one, under the SXDH assumption, as it only requires 6 elements for the signature and two pairing product equations for verification. It is defined as follows:

[1]We consider non-uniform adversaries to account for adversaries against CAnonize that may receive external advice from the environment, and following the literature (e.g. [32, 35].)

$KeyGen(n) \to (sk, pk)$.

$$
\begin{aligned}
sk =& (b, k_0, \vec{l}, d, e, k_1, \cdots, k_{n+4}, g^k) \\
& \text{with } b, k_0, d, e, k_1, \cdots, k_{n+4}, k \leftarrow \mathbb{Z}_q \\
& \quad \vec{l} \leftarrow \mathbb{Z}_q^n \\
pk =& (c_{01}, \cdots, c_{0n+4}, c_1, c_2) \\
& \text{with } a \leftarrow \mathbb{Z}_q, \\
& \quad (c_{01}, \cdots, c_{0n+4}) = (\hat{g}^{k_1 \cdot a}, \cdots, \hat{g}^{k_{n+4} \cdot a}) \\
& \quad c_1 = \hat{g}^{k \cdot a} \\
& \quad c_2 = \hat{g}^a
\end{aligned}
$$

$Sign(sk, \vec{\mu} \in \mathbb{G}_1^n) \to \sigma$.

$$
\begin{aligned}
& r, tag \leftarrow \mathbb{Z}_q \\
& \rho = g^r, \tilde{\rho} = g^{br}, \psi = g^{tag \cdot r}, \tau = \hat{g}^{tag} \\
& \gamma = (\prod_{i=1}^{n} \mu_i^{\ell_i}) \cdot g^{k_0 + d \cdot r + tag \cdot e \cdot r} \\
& \pi = \prod_{i=1}^{n} \mu_i^{k_i} \cdot \rho^{k_{n+1}} \cdot \tilde{\rho}^{k_{n+2}} \cdot \psi^{k_{n+3}} \cdot \gamma^{k_{n+4}} \cdot g^k \\
& \sigma = (\rho, \tilde{\rho}, \psi, \gamma, \tau, \pi)
\end{aligned}
$$

$Verify(pk, \sigma, \vec{\mu}) \to 0/1$.

(1) $\prod_{i=1}^{n} e(\mu_i, c_{0i}) \cdot e(\rho, c_{0n+1}) \cdot e(\tilde{\rho}, c_{0n+2}) \cdot e(\psi, c_{0n+3}) \cdot e(\gamma, c_{0n+4}) \cdot e(g, c_1) \stackrel{?}{=} e(\pi, c_2)$

(2) $e(\rho, \tau) \stackrel{?}{=} e(\psi, \hat{g})$

This structure-preserving signature is EUF-CMA secure, provided the SXDH assumption holds.

3.4 Groth–Sahai Proofs

Groth–Sahai proofs [32] enable proving the knowledge of witnesses such that statements about pairing-product equations, multi-scalar equations, and quadratic equations hold non-interactively in a witness-indistinguishable manner. The Groth–Sahai proofs follow the commit-and-prove structure: first, commitments on the witnesses are produced using a random coin, and then corresponding proof elements are computed using the same random coin (only proof elements for quadratic equations require the use of supplementary fresh randomness; see Appendix A).

Here, we are interested in the cases where we want to prove that pairing product equations of the form $e(X, \hat{B}) = T$ (type A) and $e(X, \hat{B})e(A, \hat{Y}) = T$ (type B) and multi-scalar equations $A^y = T$ (type C) hold, with $A, X \in \mathbb{G}_1, \hat{B}, \hat{Y} \in \mathbb{G}_2, y \in \mathbb{Z}_q, T \in \mathbb{G}_T$ and $X, \hat{Y}, y$ being the witnesses.

The Groth–Sahai proof system for relation R consists of three probabilistic polynomial time algorithms: a common reference string generation algorithm CRSGen taking public parameters pp as input and outputting a CRS, a prover GenProof taking as input (pp, CRS, x, w), with x the statement and w the witness and producing a proof π and a verifier VerifyProof taking as input (pp, CRS, x, π) and outputting 1 if the proof is accepted and 0 if it is rejected.

3.4.1 CRS Generation. Depending on the needs, the CRS is of the form $(\vec{g_1}, \vec{g_2})$ or $(\vec{\hat{g}_1}, \vec{\hat{g}_2})$ where $\vec{g_1}, \vec{g_2} \in \mathbb{G}_1^2$ constitute commitment

keys for the group $\mathbb{G}_1$ and $\vec{\hat{g}_1}, \vec{\hat{g}_2} \in \mathbb{G}_2^2$ constitute commitment keys for the group $\mathbb{G}_2$.

The Groth–Sahai proofs support two types of CRS distributions: a perfectly binding one and a perfectly hiding one. Those two distributions are computationally indistinguishable under the SXDH assumption [32]. Under SXDH, when committing to group elements, linearly independent CRS vectors produce perfectly hiding commitments and perfectly witness-indistinguishable proofs. On the other hand, linearly dependent CRS vectors produce perfectly binding commitments and perfectly sound proofs. When committing to an exponent $x \in \mathbb{Z}_q$, linearly independent CRS vectors can be generated to produce perfectly binding commitments to $\hat{g}^x$ and perfectly sound proofs, while linearly dependent CRS vectors produce perfectly hiding commitments and perfectly witness-indistinguishable proofs.

Groth–Sahai proofs instantiated under a binding CRS produce perfectly binding commitments that uniquely determine the committed value x if $x \in \mathbb{G}_1$ or $\mathbb{G}_2$ or to $\hat{g}^x$ if $x \in \mathbb{Z}_q$, which can be computed by a reduction holding the CRS trapdoor. We refer to this property as extractability.

3.4.2 Proof Generation and Verification. Groth–Sahai proofs constitute a commit-and-prove system, where the prover first commits to the witnesses and then produces proof elements. The commitments on values in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{Z}_q$ are defined in table 5 in Appendix A. The proof elements for the different types of equations are defined in table 6 in Appendix A. For each type of equation, the verification equations verifying the validity of the given proof are given in table 7 in Appendix A.

3.4.3 Cost. For each type of equation, the size of the proofs is presented in table 1 in terms of the number of elements in $\mathbb{G}_1$ and $\mathbb{G}_2$. The table also shows the computational cost of each type of proof, in terms of the number of exponentiations and the number of multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$.

Table 1: Size as number of elements in $\mathbb{G}_1, \mathbb{G}_2$ and number of operations of the proofs for the three types of equations

	# el. $\mathbb{G}_1$	# el. $\mathbb{G}_2$	Exp. $\mathbb{G}_1$	Mult. $\mathbb{G}_1$	Exp. $\mathbb{G}_2$	Mult. $\mathbb{G}_2$
$e(X, \hat{B}) = T$	2	2	4	3	2	
$e(X, \hat{B}) \cdot$						
$e(A, \hat{Y}) = T$	6	6	14	9	14	9
$A^y = T$	1	2	1		5	3

3.4.4 Security Guarantees. The Groth–Sahai proof protocol is a non-interactive witness-indistinguishable proof with perfect completeness, perfect soundness, and composable witness-indistinguishability for the satisfiability of a set of equations over a bilinear group where the SXDH problem is hard. Note that in the case of multi-scalar equations, zero-knowledge can be obtained in addition to witness-indistinguishability.

Composable witness-indistinguishability means that a simulated CRS is computationally indistinguishable from a real CRS and when the CRS is simulated, there is no information to distinguish the different witnesses that have been used for constructing the proofs.

Definition 3 (Perfect completeness)**.** *We say that* (*CRSGen, GenProof, VerifyProof*) *is* perfectly complete *if for all adversaries A we have*

$$\Pr\left[\begin{array}{l} CRS \leftarrow CRSGen(n, pp); \\ (x, w) \leftarrow A(pp, CRS); \\ \pi \leftarrow GenProof(pp, CRS, x, w) \end{array} : \begin{array}{l} VerifyProof(pp, CRS, x, \pi) = 1 \\ if\ (pp, x, w) \in R \end{array}\right] = 1.$$

Definition 4 (Perfect soundness)**.** *We say that* (*CRSGen, GenProof, VerifyProof*) *is* perfectly sound *if for all adversaries A we have*

$$\Pr\left[\begin{array}{l} CRS \leftarrow CRSGen(n, pp); \\ (x, \pi) \leftarrow A(pp, CRS) \end{array} : \begin{array}{l} VerifyProof(pp, CRS, x, \pi) = 0 \\ if\ \nexists w\ such\ that\ (pp, x, w) \in R \end{array}\right] = 1.$$

Definition 5 (Composable witness-indistinguishability)**.** *We say* (*CRSGen, GenProof, VerifyProof*) *is composable witness-indistinguishable, if there is a probabilistic polynomial time simulator S, such that for all non-uniform polynomial time adversaries A we have*

$$\Pr\left[CRS \leftarrow CRSGen(n, pp); : A(pp, CRS) = 1\right]$$
$$\approx \Pr\left[CRS \leftarrow S(n, pp) \quad : A(pp, CRS) = 1\right],$$

and for all adversaries A we have

$$\Pr\left[\begin{array}{l} CRS \leftarrow S(n, pp); \\ (x, w_0, w_1) \leftarrow A(pp, CRS); \\ \pi \leftarrow GenProof(pp, CRS, x, w_0) \end{array} : A(\pi) = 1\right]$$
$$= \Pr\left[\begin{array}{l} CRS \leftarrow S(n, pp); \\ (x, w_0, w_1) \leftarrow A(pp, CRS); \\ \pi \leftarrow GenProof(pp, CRS, x, w_1) \end{array} : A(\pi) = 1\right],$$

where we require $(pp, x, w_0), (pp, x, w_1) \in R$.

3.5 Naor–Pinkas–Reingold PRF

The function $f_k^H(x) := H(x)^k$, where $H : \{0,1\}^* \to \mathbb{G}$, is a PRF under the idealized assumption that H behaves as a random oracle and that DDH is hard in $\mathbb{G}$ [45].

3.6 One-Time Signature

One-time signature schemes are digital signature schemes where each key pair must be used exactly once. We use the following one-time signature, for $g, h \leftarrow \mathbb{G}_1$:

OKeyGen(n) :

$$osk = (a, b, c, d) \text{ with } a, b, c, d \leftarrow \mathbb{Z}_q$$
$$ovk = (g^a h^b, g^c h^d)$$

OSign$(osk, m \in \mathbb{Z}_q)$:

$$\sigma = (a + cm, b + dm)$$

OVerify(ovk, σ, m) :

$$ovk_1 \cdot ovk_2^m \stackrel{?}{=} g^{\sigma_1} \cdot h^{\sigma_2}$$

To the best of our knowledge, this one-time signature is one of the shortest whose security relies on an assumption implied by SXDH. Here, both the size of the one-time public key and the signature matter. This one-time signature is inspired by Groth's work [31]. While for a single use of the scheme our public key and our signature have the same size, we can reuse a generator h across many key generations so that the ephemeral one-time public key eventually

consists of two group elements (saving one). Moreover, and unlike Groth, our signing algorithm is deterministic and does not need to divide scalars. Nevertheless, those advantages do not have a big impact on the whole survey (in space and time).

Definition 6 (SUF-1CMA)**.** *We say that a one-time signature scheme* $(OKeyGen, OSign, OVerify)$ *is strongly unforgeable under a one-time chosen-message attack if for all probabilistic polynomial-time adversaries* A*, the following probability is negligible in the security parameter* n*:*

$$\Pr\left[\begin{array}{l}(\mathsf{osk},\mathsf{ovk}) \leftarrow OKeyGen(n);\\ m_0 \leftarrow A(\mathsf{ovk});\\ \sigma_0 \leftarrow OSign(\mathsf{osk}, m_0);\\ (m_1,\sigma_1) \leftarrow A(\sigma_0);\\ (m_1,\sigma_1) \neq (m_0,\sigma_0) \ \wedge \ OVerify(\mathsf{ovk},\sigma_1,m_1) = 1\end{array}\right].$$

The one-time signature scheme described above is SUF-1CMA secure under the SXDH assumption.

Indeed, the signature defines two linear functions of the message with the secret key as coefficients. A signature on one message reveals one point on each line. Producing a valid signature on a second, distinct message reveals a second independent evaluation. Two evaluations uniquely determine the underlying linear functions and hence the entire signing key.

In addition, an adversary who is able to forge a fresh signature $\sigma_1 = (\sigma_{1,1}, \sigma_{1,2})$ on a message m that has already been signed with signature $\sigma_0 = (\sigma_{0,1}, \sigma_{0,2})$, can be used to solve the DDH problem in $\mathbb{G}_1$. Indeed, as σ_0, σ_1 are both valid, they will both pass the verification equation, which means we have $\mathsf{ovk}_1 \cdot {\mathsf{ovk}_2}^m = g^{\sigma_{1,1}} h^{\sigma_{1,2}} = g^{\sigma_{0,1}} h^{\sigma_{0,2}}$. From which we can derive the discrete logarithm of h in base g as $h = g^{\frac{\sigma_{1,1}-\sigma_{0,1}}{\sigma_{0,2}-\sigma_{1,2}}}$, which contradicts the DDH assumption in $\mathbb{G}_1$.

3.7 Fischlin Transform

The Fischlin transform [21, 28] enables turning a three-move proof of knowledge $(\mathbb{P}, \mathbb{V})$ into a non-interactive one $(\mathbb{P}', \mathbb{V}')$ while providing straight-line extraction in the random oracle model. Straight-line extraction means that the witnesses of the proof are extracted instantaneously, without requiring rewinding or forking, which can result in substantial loss in security reductions [13, 53]. Following the terminology of Chen and Lindell [21], and based on the security parameters $\rho = 32$, $b = 4$ and $t = 9$ chosen to reach a 128-bit security level, the completeness error can be bounded by $\epsilon_{comp} \leq \rho \cdot e^{-2^{t-b}}$ and the soundness error can be bounded by $\epsilon_{sound} \leq (Q+1) \cdot 2^{-\rho \cdot b}$, where Q is the number of queries made by the adversary to the random oracle.

4 Anonymous Survey Scheme

In this section, we present the anonymous survey security model of Hohenberger et al., which they call *ad-hoc surveys* [35], defining its participants and protocol blueprint (Sec. 4.1) and the desired security properties (Sec 4.2).

4.1 Ad-Hoc Survey Scheme

An ad-hoc survey scheme involves three types of entities: a single registration authority (RA), survey authorities (SA), and users. The RA is responsible for maintaining a list of registered users. Any user wanting to organize a survey can assume the role of the SA. Each user is identified by their public id.

The entities execute the following protocols, whose role we detail below.

Definition 7. *An ad-hoc anonymous survey scheme* Γ *consists of the following PPT protocols and algorithms:*

- $\mathrm{Setup}(n) \rightarrow pp$*, which produces the public parameters made available to all the other protocols,*
- $\mathrm{GenRA}(n) \rightarrow sk_{\mathrm{RA}}, pk_{\mathrm{RA}}$*, the registration authority key generation algorithm, with security parameter* n*,*
- $\mathrm{UR}^{\mathrm{RA}}(n, sk_{\mathrm{RA}}, id) \rightarrow$ `accept/fail` *(interactive), the user registration protocol executed by the registration authority,*
- $\mathrm{UR}^{\mathrm{U}}(n, id, pk_{\mathrm{RA}}) \rightarrow cred_{id}/$`fail` *(interactive), the user registration protocol executed by the user,*
- $\mathrm{GenSA}(n) \rightarrow sk_{\mathrm{SA}}, pk_{\mathrm{SA}}$*, the survey authority key generation algorithm,*
- $\mathrm{SR}(n, sk_{SA}, vid, L) \rightarrow L_{vid}$*, the survey registration protocol executed by any survey authority producing a list of survey credentials,*
- $\mathrm{Auth}(n, vid, L_{vid}, id, pk_{\mathrm{SA}}) \rightarrow$ `accept/fail`*, the survey credential verification protocol,*
- $\mathrm{Submission}(n, m, cred_{id}, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}}, L_{vid}[id]) \rightarrow Sub = (C, m, \pi)$*, the survey response submission protocol,*
- $\mathrm{CheckSubmission}(n, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}}, Sub) \rightarrow$ `accept/fail`*, the survey response verification protocol.*

System Setup. The Setup process generates the required public parameters, including group descriptions. For systems that rely on a Common Reference String (CRS), the CRS is generated and made publicly available. For notational convenience, we omit the public parameters and CRS from the explicit input of the protocols and assume it is provided implicitly. Similarly, when working in the Random Oracle model (ROM), we assume that all protocols have implicit access to the random oracle.

During the setup phase, the registration authority also generates a key pair using GenRA and makes its public key available.

User Registration. The user sets up a mutually authenticated secure connection with the RA, and through the UR^{U} and $\mathrm{UR}^{\mathrm{RA}}$ protocols, the user obtains fresh credentials $cred_{id}$ if he has not run the registration process yet.

Survey Registration. The SA chooses an identifier vid for the survey it creates and a list L of authorized participants to the survey. The SA generates a survey key pair using GenSA and publishes the corresponding public key, and also generates a list of survey credentials L_{vid} corresponding to the users in L, using SR.

Authorization Check. To prevent a survey from being targeted at deanonymizing a particular user, anyone can use Auth to verify whether a user with a specific id is authorized to participate by checking the validity of the corresponding survey credential in the L_{vid} list published by the SA. We are interested in cases where the set of authorized users is naturally interpretable (e.g., email addresses with a specific domain).

Survey Participation. The user and the SA establish a secure channel where the SA is authenticated. The user uses Submission

to submit to a specific survey a response m, a proof π showing that he is registered and authorized, and a token C. This token is unlinkable to the user identity id, and its value is unique for a given user and a given survey, thus ensuring single participation. If the validity of the submission is verified using CheckSubmission, the SA records it, replacing any previous submission containing the same token.

Bulletin Board. A bulletin board could enable users to check whether their answer is correctly registered and could also enable third parties to verify response rates and submission validity via the public procedure CheckSubmission. However, we do not define it formally or explore this possibility any further here. Instead, we focus on SAs that genuinely desire to collect and learn from survey responses: in that setting, SAs serve as designated verifiers responsible for collecting submissions, though they may still be interested in breaking anonymity/unlinkability.

4.2 Desired properties

An anonymous survey scheme should satisfy the three following properties: correctness, unlinkability, and security against malicious users. We use the same definitions as in [35], which we recall here.

4.2.1 Correctness. For each survey set up by an SA for a list of users L, if a user in L has correctly registered with the RA, then the user should be authorized to submit a response that passes the check. We also require that all authorized users' submissions have different token numbers.

Definition 8. *An ad-hoc survey scheme Γ is correct if there exists a negligible function $\mu(\cdot)$, such that the following experiment outputs* `fail` *with probability at most $\mu(n)$ for every $n \in \mathbb{N}$, vid, set L of n-bit strings, every mapping $m : L \to \{0,1\}^n$:*

- $(sk_{\mathrm{RA}}, pk_{\mathrm{RA}}) \leftarrow \mathrm{GenRA}(n)$
- $(sk_{SA}, pk_{SA}) \leftarrow \mathrm{GenSA}(n)$
- *For every $id \in L$:*
  - $(out_{RA}, out_U) \leftarrow out[\mathrm{UR}^{\mathrm{RA}}(n, sk_{\mathrm{RA}}, id) \leftrightarrow \mathrm{UR}^{\mathrm{U}}(n, id, pk_{\mathrm{RA}})]$
  - *Output* `fail` *if either out_{RA} or out_U is* `fail`*; otherwise let $cred_{id} = out_U$.*
- $L_{vid} \leftarrow \mathrm{SR}(n, sk_{SA}, vid, L)$
- *Output* `fail` *if there exists some $id \in L$ such that* $\mathrm{Auth}(n, vid, L_{vid}, id, pk_{\mathrm{SA}}) = \texttt{fail}$.
- *For every $id \in L$:*
  - $Sub_{id} = (C_{id}, m_{id}, \pi_{id}) \leftarrow \mathrm{Submission}(n, m[id], cred_{id}, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}}, L_{vid}[id])$.
  - *Output* `fail` *if* $\mathrm{CheckSubmission}(n, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}}, Sub_{id}) = \texttt{fail}$ *or if $m_{id} \neq m[id]$.*
- *Output* `fail` *if there exists distinct $id, id' \in L$ such that $C_{id} = C_{id'}$ for $Sub_{id'} = (C_{id'}, m_{id'}, \pi_{id'})$.*

4.2.2 Unlinkability. Nobody, including the other users, the RA and the SAs, in collusion or not, should be able to link a specific submission to a specific user, even if the link between the user and submission in other surveys is revealed to the attacker.

Definition 9. *An ad-hoc survey scheme Γ is unlinkable if for every non-uniform polynomial-time adversary A, the ensembles $\{\mathrm{EXEC}^0(n, A)\}_{n\in\mathbb{N}}$ and $\{\mathrm{EXEC}^1(n, A)\}_{n\in\mathbb{N}}$ are computationally indistinguishable.* $\mathrm{EXEC}^b(n, A)$ *is defined as:*

- $(pk_{\mathrm{RA}}, z) \leftarrow A(n)$
- *Let adversary $A(n, z)$ concurrently interact with $\mathrm{UR}^{\mathrm{U}}(n, \cdot, \cdot)$ with adaptively chosen, but unique, user identities $id \in \{0,1\}^n$ of its choice. That is, A can only use each user identity id in a single interaction with UR^{U}. Whenever an interaction with some $\mathrm{UR}^{\mathrm{U}}(n, id, pk_{\mathrm{RA}})$ successfully completes (i.e., the output of the user is not* `fail`*), let $cred_{id}$ be the output of UR^{U}, and for the remainder of the experiment, A gets oracle access to* $\mathrm{Submission}(n, \cdot, cred_{id}, \cdot, \cdot, \cdot, \cdot)$.
- $(vid, pk_{\mathrm{SA}}, L_{vid}, id_0, id_1, m_0, m_1, z') \leftarrow A(n, z)$
- *Output* `fail` *if* $\mathrm{Auth}(n, vid, L_{vid}, id_\beta, pk_{\mathrm{SA}}) = \texttt{fail}$ *for any $\beta \in \{0,1\}$.*
- *Let* $Sub_\beta = \mathrm{Submission}(n, m_\beta, cred_{id_{\beta\oplus b}}, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}}, L_{vid}[id_{\beta\oplus b}])$.
- $out \leftarrow A(n, (Sub_0, Sub_1), z')$
- *Output* `fail` *if there exists some $\beta \in \{0,1\}$ such that A queried its* $\mathrm{Submission}(n, \cdot, cred_{id_\beta}, \cdot, \cdot, \cdot, \cdot)$ *oracle with vid as its fifth input; otherwise, output out.*

This definition considers an RA and an SA under adversarial control, but we stress out that it only guarantees unlinkability for users submitting a single response per vid, and says nothing if a user decides to submit two responses to a single survey, or if a malicious SA creates a survey with a vid copied from another survey. Using "one-time" zero-knowledge techniques as for double-spending detection in e-cash [19] or duplicate signature detections in linkable signatures [9], the user identity could be revealed if the same user submits two responses with the same vid. This is also the case for the generic anonymous survey construction presented in Anonize. However, neither their concrete instantiation nor our construction suffers from this issue. In both cases, submitting two responses with the same vid only reveals that they originate from the same user, without revealing the user's identity.

4.2.3 Security Against Malicious Users. A user can complete each survey for which they are authorized at most once. In this way, we avoid malicious users trying to influence the result of the survey by answering multiple times.

Definition 10. *An ad-hoc survey scheme Γ is secure against malicious users if for every non-uniform polynomial-time A, every polynomial $p(\cdot)$, there exists a negligible function $\mu(\cdot)$, such that the following experiment outputs accept with probability at most $\mu(n)$ for every $n \in \mathbb{N}$:*

- $(sk_{\mathrm{RA}}, pk_{\mathrm{RA}}) \leftarrow \mathrm{GenRA}(n)$
- $(sk_{SA}, pk_{SA}) \leftarrow \mathrm{GenSA}(n)$
- *Honest user registration and survey submission: Throughout the experiment, $A(n)$ has oracle access to* $\mathrm{Submission}'(n, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot, sk_{\mathrm{RA}})$. *The* $\mathrm{Submission}'(n, m, id, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}}, L_{vid}, sk_{\mathrm{RA}})$ *oracle outputs, if the oracle has not previously been queried on the identity id, the output of* $\mathrm{Submission}(n, m, cred_{id}, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}}, L_{vid}[id])$ *with $(out_{RA}, cred_{id})$, the result of an execution of $\mathrm{UR}^{\mathrm{RA}}(n, sk_{\mathrm{RA}}, id) \leftrightarrow \mathrm{UR}^{\mathrm{U}}(n, id, pk_{\mathrm{RA}})$; otherwise, it recovers the previously computed credential $cred_{id}$ and outputs* $\mathrm{Submission}(n, m, cred_{id}, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}},$

$L_{vid}[id]$). *Let* L_{honest} *be the set of identities on which A queries the oracle (as its seventh input).*

- *Corrupt user registration: Throughout the experiment, A can concurrently interact with* $\mathrm{UR}^{\mathrm{RA}}(n, sk_{\mathrm{RA}}, \cdot)$ *with adaptively chosen, but unique, user identities* $id \in \{0,1\}^n$ *of its choice. That is, A can only use each user identity id in a single interaction with* $\mathrm{UR}^{\mathrm{RA}}$. *Let* $L_{corrupt}$ *be the set of identities on which* $\mathrm{UR}^{\mathrm{RA}}$ *outputs* `accept`.
- $z, L, vid \leftarrow A(n)$
- $L_{vid} \leftarrow \mathrm{SR}(n, sk_{SA}, vid, L)$
- $S \leftarrow A(n, z, pk_{\mathrm{SA}}, L_{vid})$
- *Output* `accept` *if:*
  (1) $L_{honest} \cap L_{corrupt} = \emptyset$
  (2) $|S| > |L \cap L_{corrupt}|$
  (3) *For all* $Sub \in S$, $\mathrm{CheckSubmission}(n, pk_{\mathrm{RA}}, vid, pk_{\mathrm{SA}}, Sub)$ *accepts*
  (4) *For any two* $(C, m, \pi), (C', m', \pi') \in S$, $C \neq C'$
  (5) *For all* $(C, m, \pi) \in S$, $\nexists (C^*, m^*, \pi^*)$ *output by* $\mathrm{Submission}'(n, m, \cdot, vid, \cdot, \cdot)$ *such that* $(C, m) = (C^*, m^*)$.

In the acceptance conditions, Conditions (2), (3), (4) and (5) together require that S contains more adversarially generated submissions (Cond. 5) than the number of corrupted users (Cond. 2), and that these submissions must be considered valid (Cond. 3) and have a unique token (Cond. 4).

This definition considers an honest RA and allows the presence of malicious SAs. It guarantees that an honest SA will only accept at most one submission per user and per vid, and this remains true even if multiple surveys with the same *vid* are created by other (potentially malicious) SAs.

5 Proposed Survey Scheme

In this section, we present CAnonize. The scheme follows the high-level structure of Anonize [35] but modifies its core cryptographic components in order to (i) simplify the underlying assumptions and (ii) improve computational and communication efficiency, while preserving the unlinkability, correctness, and security against malicious users properties.

Concretely, we replace: (i) the generic NIZK proof system with Groth–Sahai (GS) proofs, which significantly reduces the number of pairing computations required from the user, (ii) partially blind signatures with a structure-preserving signature (SPS) scheme that accommodates efficient GS-based verification, and (iii) the Dodis–Yampolskiy PRF with a Naor–Pinkas–Reingold PRF, which allows the token to be in $\mathbb{G}_1$ instead of $\mathbb{G}_T$. The combination of the Groth–Sahai proofs with the structure-preserving signature scheme removes the need for rewinding in security proofs, which was required for the proof system involving a blind signature scheme used in Anonize. Additionally, the Naor–Pinkas–Reingold PRF does not provide a self-verifiability property as induced by the Dodis–Yampolskiy PRF used in Anonize and thus requires the verification of a supplementary condition by the Groth–Sahai proof system. We further modify the user registration and submission steps to guarantee the security of the scheme under the new primitives.

These changes allow the scheme to be instantiated under the SXDH assumption in asymmetric bilinear groups, eliminating the need for the combination of the Discrete Logarithm hardness, the Decisional Bilinear Diffie–Hellman, and the n-Decisional Diffie–Hellman Inversion assumptions required by Anonize. Furthermore, the number of pairing evaluations required from the user is significantly reduced.

Security is achieved in the common reference string (CRS) model under the SXDH assumption, modeling hash functions by random oracles.

5.1 High-Level Description

We present a high-level description of the main steps of CAnonize and give intuition about the proofs generated during submission. Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ be asymmetric bilinear groups of prime order q with generators $g \in \mathbb{G}_1$ and $\hat{g} \in \mathbb{G}_2$.

5.1.1 User Registration. The user chooses a secret PRF seed *sid*, which will allow the user to compute an unlinkable one-time token C during submission. Unlike in Anonize, where *sid* is blindly signed by the RA, we encode it as a group element pk, which does not reveal information about *sid*, and prove that it is well-formed using Groth–Sahai proofs.

$$pk = g^{sid}$$

The unlinkable user credential is provided by the RA as the signature on (g^{id}, pk) using an SPS scheme.

5.1.2 Survey Registration. The SA signs (g^{id}, vid) for each authorized user in L with the SPS signature scheme. The SA publishes the list of survey credentials L_{vid} as a list of the user identities and their associated signatures. This per-user credential pre-issuance, assuming the SA's honesty, allows for constant cost submissions.

5.1.3 Submission. The user generates a token C, using the Naor–Pinkas–Reingold PRF:

$$C = H(vid)^{sid},$$

where H hashes to $\mathbb{G}_1$. This token is unique for each user and survey, ensuring that users can participate at most once, while hiding *sid* and thus preserving unlinkability.

Using Groth–Sahai proofs, the user proves that the following two conditions hold together:

(a) they hold σ_{RA}, a valid SPS signature from the RA on (g^{id}, pk) and $\sigma_{id,vid}$, a valid SPS signature from the SA on (g^{id}, vid);
(b) the submitted token satisfies $C = H(vid)^{sid}$, using the same value of *sid* as in pk.

The user further signs all proof elements, *vid* and the message m using a one-time signature scheme to prevent replay and proof malleability.

5.1.4 Intuition About Proofs of Conditions (a) and (b). The proof for condition (a) is constructed as follows: the user proves knowledge of the witnesses g^{id}, pk, σ_{RA}, and $\sigma_{id,vid}$ such that the verification equations for both the RA's and SA's SPS signatures hold. Since GS proofs follow a commit-and-prove structure, the user first commits to the witnesses using random coins and then constructs proof elements relative to these commitments. Specifically, a single Groth–Sahai commitment to g^{id} is used in the proof elements for both signature verification equations, guaranteeing that the same value of g^{id} is certified in both signatures and thereby ensuring that the

registration and authorization credentials correspond to the same underlying identity.

Condition (b) compensates for the loss of the self-verifiability property when replacing the Dodis–Yampolskiy PRF used in Anonize with the Naor–Pinkas–Reingold PRF. To enforce condition (b), the user proves that the token C is consistent with the value of pk, without revealing pk itself, as disclosing it would enable linkability to the user credential, where (g^{id}, pk) is certified by the RA. Concretely, the user proves that the exponent sid used in $C = H(vid)^{sid}$ is the same as the exponent underlying the commitment to $pk = g^{sid}$ generated for proving condition (a). Since pk is already committed to in the first proof, this consistency statement is instantiated under an independent CRS, that is newly generated for each submission, to ensure sound composition of the two proofs and to avoid unintended correlations between proof elements. The per-user CRS generation allows controlling the distribution of the proofs to maintain the extractability of those adversarially produced from those that need to be simulated. The per-user CRS cannot be derived directly from the user identity without breaking the unlinkability. We thus partition the CRSes thanks to ephemeral public keys of a one-time signature scheme used as a salt. Since these ephemeral salts are public, an adversary could still be tempted to reuse one from an honest user for which we would like the proof to be simulatable. To avoid that, Canonize requires users to sign their surveys with the associated ephemeral secret key. Since any valid submission needs to include such a valid signature, any malicious attempt to reuse an honest CRS also needs to come with another valid signature for the corresponding ephemeral public key, breaking its unforgeability.

Since C is deterministically derived as $H(vid)^{sid}$, a user cannot produce two distinct well-formed tokens for the same survey. Combined with server-side duplicate detection of previously submitted tokens, this guarantees at most one accepted submission per user and per survey.

Together, conditions (a) and (b) guarantee that any accepting submission corresponds to a valid credential chain issued by the RA and SA, bound to a well-formed token C. Additionally, while the user is identified by g^{id} to both the RA and SA, their identity remains anonymous during the survey submission process as the proofs reveal no information about id or sid beyond the validity of the stated relations.

5.2 Concrete Instantiation

5.2.1 Primitives Used. We describe the primitives used in CAnonize:

- The structure-preserving signature scheme with algorithms $\mathrm{KeyGen}(n) \to (sk, pk)$, $\mathrm{Sign}(sk, m) \to \sigma$, $\mathrm{Verify}(pk, m, \sigma) \to \{0,1\}$.
- The Groth–Sahai proof system for non-interactive proofs of pairing-product equations. We denote by $\mathrm{CRSGen}_{\mathbb{G}_X}(n) \to \mathrm{CRS}$ the CRS generation algorithm in $\mathbb{G}_X$ for $X \in \{1,2\}$, $\mathrm{GenProof}_Y(\mathrm{CRS}, x, w) \to \pi_Y$ the proof generation algorithms for $Y \in \{\mathrm{UR}, \sigma\}$, respectively for the proof in user registration and for the proof of condition (1) in submission, described in the high-level description, and $\mathrm{VerifyProof}_Y(\mathrm{CRS}, x, \pi_Y) \to \{0,1\}$ the proof verification algorithm for $Y \in \{\mathrm{UR}, \sigma\}$. The concrete proof elements are given in Appendix B.
- The Groth–Sahai proof system for non-interactive proofs of multi-scalar multiplication equations. We denote by $\mathrm{CRSGen}'_{\mathbb{G}_2}(n) \to \mathrm{CRS}'[2]$ the algorithm generating the second vector of the commitment key for $\mathbb{G}_2$, $\mathrm{GenProof}_C(\mathrm{CRS}, x, w) \to \pi_C$ the proof generation algorithm for the proof of condition (2) in submission and $\mathrm{VerifyProof}_C(\mathrm{CRS}, x, \pi_C) \to \{0,1\}$ the corresponding proof verification algorithm. The concrete proof elements are also given in Appendix B.
- The one-time signature scheme $\mathrm{OKeyGen}(n) \to (osk, ovk)$, $\mathrm{OSign}(osk, m) \to \sigma_O$, $\mathrm{OVerify}(ovk, m, \sigma_O) \to \{0,1\}$.
- The hash-to-curve functions $H : \{0,1\}^* \to \mathbb{G}_1$, $H' : \{0,1\}^* \times \{0,1\}^* \to \mathbb{G}_2$ and collision-resistant hash function $H'' : \{0,1\}^* \to \mathbb{Z}_q$.

For the purpose of evaluating CAnonize, we instantiate it with concrete primitives. In principle, any structure-preserving signature can be used by adapting the verification equations in the Groth–Sahai proofs. Similarly, any one-time signature scheme, hash function, or hash-to-curve function can be substituted as needed. However, the usage of the Groth–Sahai proofs is not black-box, as we need to make a proof in condition (b) on the commitment on pk generated in condition (a). Additionally, some parts of the CRS are generated on-the-fly for each submission, and this partial CRS generation relies on the specific structure of the Groth–Sahai CRS.

5.2.2 Construction. An anonymous survey scheme consists of the following algorithms and is described in figure 1.

Setup. Generates public parameters pp describing $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ with random generators $g, h \in \mathbb{G}_1$ and $\hat{g} \in \mathbb{G}_2$.

GenRA and GenSA. The RA and SA respectively generate SPS key pairs:

$$(sk_{RA}, pk_{RA}), (sk_{SA}, pk_{SA}).$$

The RA also creates an empty list to store registered users, and the SA sets the survey identifier vid and the list of authorized users L.

CRS Generation. Generates Groth–Sahai commitment keys:

- $\mathrm{CRS}_{\mathrm{UR}}$ for ensuring the knowledge of $\hat{g}^{sid}$ during user registration;
- CRS_1 and CRS_2 for proving possession of valid SPS signatures;
- CRS' for proving correctness of token generation.

Let us denote the first and second vectors of CRS' as $\mathrm{CRS}'[1]$ and $\mathrm{CRS}'[2]$. Only $\mathrm{CRS}'[2]$ is generated during setup, and $\mathrm{CRS}'[1]$ is computed during submission using a hash of the one-time verification key and the survey identifier.

User Registration (UR). The user and the RA establish a mutually authenticated secure connection. The user with identity $id \in \mathbb{Z}_q$:

(1) Samples $sid \leftarrow \mathbb{Z}_q$ and computes $pk = g^{sid}$.
(2) Generates a GS proof π_{UR} ensuring knowledge of $\hat{g}^{sid}$ such that $pk = g^{sid}$.
(3) Sends $(g^{id}, pk, \pi_{\mathrm{UR}})$ to the RA.

If the proof verifies and id is not yet registered, the RA signs (g^{id}, pk) using the SPS scheme and returns σ_{RA}.

The user stores credentials $cred_{id} = (g^{id}, pk, \sigma_{\mathrm{RA}}, sid)$.

Survey Registration (SR).. For survey identifier vid encoded in $\mathbb{G}_1$ and participant list L, the SA publishes the authorization list:

$$L_{vid} = \{(id, \sigma_{id,vid})\}_{id \in L}, \text{ where } \sigma_{id,vid} \leftarrow \text{Sign}(sk_{SA}, (g^{id}, vid)).$$

Authorization Check. To check if a user with identity id is authorized for survey vid, anyone can verify the presence of a valid signature $\sigma_{id,vid}$ in L_{vid} using pk_{SA}.

Submission. The user and the SA establish a secure channel where only the SA is authenticated. To submit a response m, the user holding $cred_{id}$ proceeds as follows:

(1) Computes the token $C = H(vid)^{sid}$.
(2) Generates a fresh one-time signature key pair (osk, ovk).
(3) Completes CRS′ by setting $\text{CRS}'[1] = H'(ovk, vid)$.
(4) Generates Groth–Sahai proofs π_σ on pairing product equations and π_C on multi-scalar multiplication equations:
   (a) π_σ: σ_{RA} is a valid signature on (g^{id}, pk) with pk_{RA} and $\sigma_{id,vid}$ is a valid signature on (g^{id}, vid) with pk_{SA}, using $\text{CRS}_1, \text{CRS}_2$.
   π_σ contains a commitment c_{pk} to pk, using randomness r_1^{pk}, r_2^{pk}.
   (b) π_C: c_{pk} and $C = H(vid)^{sid}$ share the same sid using CRS′.
(5) Computes a one-time signature binding all the proof elements, the token, and the survey identifier to the message:

$$\sigma_O = Sign_{osk}(H''(\pi_\sigma, \pi_C, C, vid, m)).$$

The submission is:

$$Sub = (C, m, \pi_\sigma, \pi_C, \sigma_O, ovk).$$

Submission Verification. Upon receiving Sub, the SA:

(1) Recomputes $\text{CRS}'[1] = H'(ovk, vid)$.
(2) Verifies π_σ, π_C.
(3) Recomputes the hash $H''(\pi_\sigma, \pi_C, C, vid, m)$ and verifies the one-time signature σ_O.

If all checks pass and C has not appeared previously for this survey, the submission is accepted, and C is recorded.

Please refer to Appendix B for the concrete proof elements and verification equations for π_{UR}, π_σ, and π_C.

5.3 Security

The ad-hoc survey scheme relies on the structure-preserving signature, the Groth–Sahai proofs, the Naor–Pinkas–Reingold pseudo-random function, and the one-time signature. We assume these schemes are secure when instantiated with the group generator $\mathcal{G}$ in the ROM and CRS models. The structure-preserving signature, the Groth–Sahai proofs, and the one-time signature rely on the assumption of the hardness of the SXDH problem, while the Naor–Pinkas–Reingold pseudo-random function relies on the hardness of the DDH problem, which is guaranteed by the hardness of the SXDH problem. The ROM is used for the Naor–Pinkas–Reingold pseudo-random function and for the hash-to-curve function H', while the CRS model is used for the Groth–Sahai proofs. So, CAnonize only relies on the hardness of the SXDH problem in the ROM and in the CRS model, and on the security of the primitives used:

$\text{Setup}(n)$
return $\text{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \hat{g})$

$\text{GenRA}(n)$
$L_{users} = \emptyset$
$sk_{\text{RA}}, pk_{\text{RA}} \leftarrow \text{KeyGen}(n)$
return $sk_{\text{RA}}, pk_{\text{RA}}$

$\text{GenSA}(n)$
$sk_{\text{SA}}, pk_{\text{SA}} \leftarrow \text{KeyGen}(n)$
return $sk_{\text{SA}}, pk_{\text{SA}}$

$\text{GenCRS}(n)$
$\text{CRS}_{\text{UR}} \leftarrow \text{CRSGen}_{\mathbb{G}_2}(n)$
$\text{CRS}_1 \leftarrow \text{CRSGen}_{\mathbb{G}_1}(n)$
$\text{CRS}_2 \leftarrow \text{CRSGen}_{\mathbb{G}_2}(n)$
$\text{CRS}'[2] \leftarrow \text{CRSGen}'_{\mathbb{G}_2}(n)$
(only the 2^{nd} element of the CRS)
return $\text{CRS}_{\text{UR}}, \text{CRS}_1, \text{CRS}_2, \text{CRS}'[2]$

$\text{UR1}^U(n, id)$
$sid \leftarrow \mathbb{Z}_q$
$pk = g^{sid}$
$\pi_{\text{UR}} \leftarrow \text{GenProof}_{\text{UR}}(n, \text{CRS}_{\text{UR}}, sid)$
send id, pk, π_{UR} to RA

$\text{UR}^{RA}(n, sk_{\text{RA}}, id, pk, \pi_{\text{UR}})$
if $\text{VerifyProof}_{\text{UR}}(n, \text{CRS}_{\text{UR}}, \pi_{\text{UR}}) = 1$
$\wedge$ id not registered:
  add id to L_{users}
  $\sigma_{\text{RA}} = \text{Sign}(sk_{\text{RA}}, (g^{id}, pk))$
  send σ_{RA} to user and **return** `accept`
else return `fail`

$\text{UR2}^U(n, \sigma_{RA}, pk_{\text{RA}})$
if $\text{Verify}(n, pk_{\text{RA}}, g^{id}, pk, \sigma_{RA}) \stackrel{?}{=} 1$:
  return $cred_{id} = (g^{id}, pk, \sigma_{RA}, sid)$
else return `fail`

$\text{SR}(n, sk_{\text{SA}}, vid, L)$
for each $id \in L$:
  $\sigma_{id,vid} \leftarrow \text{Sign}(n, sk_{\text{SA}}, (g^{id}, vid))$
return $L_{vid} = \{(id, \sigma_{id,vid})\}_{id \in L}$

$\text{Auth}(n, vid, L_{vid}, id, pk_{\text{SA}})$
if $\exists (id, \sigma_{id,vid}) \in L_{vid}$ s.t.
$\text{Verify}(n, pk_{SA}, g^{id}, vid, \sigma_{id,vid}) = 1$
  return `accept`
else return `fail`

$\text{Submission}(n, m, cred_{id}, pk_{\text{RA}}, vid, pk_{\text{SA}}, \sigma_{id,vid})$
$C = H(vid)^{sid}$
$(osk, ovk) \leftarrow \text{OKeyGen}(n)$
$\text{CRS}'[1] \leftarrow H'(ovk, vid)$
$\text{CRS}' := (\text{CRS}'[1], \text{CRS}'[2])$
$\pi_\sigma \leftarrow \text{GenProof}_\sigma(n, (\text{CRS}_1, \text{CRS}_2), (pk_{\text{RA}}, pk_{\text{SA}}), (g^{id}, pk, \sigma_{\text{RA}}, \sigma_{id,vid}))$
$\pi_C \leftarrow \text{GenProof}_C(n, \text{CRS}', (sid, r_1^{pk}, r_2^{pk}))$
  with r_1^{pk}, r_2^{pk} the random coins used in c_{pk},
    which is contained in π_σ
$\sigma_O \leftarrow \text{OSign}(n, osk, H''(\pi_\sigma, \pi_C, C, vid, m))$
return $Sub = (C, m, \pi_\sigma, \pi_C, \sigma_O, ovk)$

$\text{CheckSubmission}(n, m, pk_{\text{RA}}, vid, pk_{\text{SA}}, Sub)$
$\text{CRS}'[1] \leftarrow H'(ovk, vid)$
$\text{CRS}' := (\text{CRS}'[1], \text{CRS}'[2])$
if $\text{VerifyProof}_\sigma(n, (\text{CRS}_1, \text{CRS}_2), (pk_{\text{RA}}, pk_{\text{SA}}), \pi_\sigma) = 1$
$\wedge$ $\text{VerifyProof}_C(n, \text{CRS}', \pi_C) = 1$
$\wedge$ $\text{OVerify}(n, ovk, H''(\pi_\sigma, \pi_C, C, vid, m)) = 1$
  return `accept`
else return `fail`

Figure 1: Different steps of the anonymous survey scheme

Assumption 1. *When instantiated with group generator $\mathcal{G}(\lambda)$, the structure-preserving signature (Section 3.3) is an unbounded CMA-secure signature scheme.*

Assumption 2. *When instantiated with group generator $\mathcal{G}(\lambda)$ in the hiding mode, the Groth–Sahai proof system (Section 3.4) is a perfectly witness-indistinguishable proof system. Moreover, in the case of multi-scalar multiplication equations, it is a zero-knowledge proof system.*

Assumption 3. *When instantiated with group generator $\mathcal{G}(\lambda)$ in the binding mode, the Groth–Sahai proof system (Section 3.4) is a perfectly sound proof system. In the case of PPE, the proof is extractable. In the case of multi-scalar multiplication equations, witnesses $w \in \mathbb{Z}_q$ can be extracted as $\hat{g}^w$.*

Assumption 4. *When instantiated with group generator $\mathcal{G}(\lambda)$, both modes of the Groth–Sahai proof system have indistinguishable CRSes.*

Assumption 5. *When instantiated with group generator $\mathcal{G}(\lambda)$, the Naor–Pinkas–Reingold function (Section 3.5) is a secure pseudo-random function that maps λ-bit strings to elements of $\mathbb{G}_1$.*

Assumption 6. *When instantiated with group generator $\mathcal{G}(\lambda)$, the one-time signature (Section 3.6) is SUF-1CMA secure.*

Theorem 1. *Under assumptions 1 to 6, CAnonize detailed in Section 5.2 is a correct, unlinkable ad-hoc survey scheme that is secure against malicious users in the random oracle model. In particular, the scheme has public-coin CRS, and all these security notions hold solely under the SXDH assumption in the ROM.*

Corollary 1.1. *For any non-uniform PPT adversary $\mathcal{A}$, making at most Q random-oracle queries to $H(\cdot)$, Q' queries to $H'(\cdot,\cdot)$, Q'' queries to $H''(\cdot,\cdot,\cdot,\cdot,\cdot)$, and let ϵ_{SXDH} be the advantage against the SXDH problem and ϵ_{SPS} the advantage against the structure-preserving signature scheme, we denote by $\mathsf{Adv}_\lambda^{\text{unlink.}}(n)$ the advantage of $\mathcal{A}$ in the unlinkabilitiy game and $\mathsf{Adv}_\lambda^{\text{security}}(n)$, the advantage of $\mathcal{A}$ in the security against malicious users game.*

Then the following bounds hold:

$$\mathsf{Adv}_\lambda^{\text{unlink.}}(n) \leq (Q+2)\cdot\epsilon_{\text{SXDH}}$$

and

$$\mathsf{Adv}_\lambda^{\text{security}}(n) \leq 7\cdot\epsilon_{\text{SXDH}} + \frac{Q'^2}{q^2} + \frac{Q''^2}{q} + 2\cdot\epsilon_{SPS}$$

Proof.

Correctness. Assuming the security of the Groth–Sahai proofs, the correctness experiment will succeed for the user registration step. Then, assuming the security of the signature scheme, the signature verification in Auth will pass. Assuming the security of the PRF, the token C is correctly computed, and assuming the security of the hash-to-curve function H' and the hash function H'', CRS′ will have the same value in Submission and CheckSubmission, and the output of H'' will be the same in the submission and the submission checking phases. Now, assuming the security of the Groth–Sahai proofs and the one-time signature scheme, the proofs generated during submission and the one-time signature will verify. Finally, the probability that two different users have the same token numbers is negligible, as the PRF seed *sid* is taken uniformly at random. The correctness experiment will thus fail with at most negligible probability.

Unlinkability. Let A be a non-uniform PPT adversary against the unlinkability of the anonymous survey. We consider the following sequence of hybrid experiments:

- Hybrid 0 is the real unlinkability experiment $\text{EXEC}^b(n, A)$. Since the common reference string is public-coin, CRS_1 and CRS_2 are random group elements allowing to commit to group elements in a perfectly hiding manner. In that case, the GS proof π_σ is perfectly witness-indistinguishable (Assumption 2): π_σ is independent of the witness $(g^{id}, pk, \sigma_{\text{RA}})$, $\sigma_{id,vid}$ being used.
- Hybrid 1 is identical to Hybrid 0 except that instead of having CRS_{UR} form a uniform quadruple in $\mathbb{G}_2$, we set $\text{CRS}_{\text{UR}}[1] = \text{CRS}_{\text{UR}}[2]^\eta$ for a random scalar $\eta \leftarrow \mathbb{Z}_q$. This turns CRS_{UR} into hiding mode, allowing to commit to exponents in a perfectly hiding manner and allowing the GS proof π_{UR} to be zero-knowledge. π_{UR} is thus independent of the witness *sid* used in the user registration phase. By the hardness of the SXDH assumption and the zero-knowledge property of π_{UR} (Assumption 2), the output of Hybrid 1 is indistinguishable from Hybrid 0 with security loss ϵ_{SXDH} (Assumption 4), where ϵ_{SXDH} is the SXDH advantage.
- Hybrid 2 is identical to Hybrid 1 except that, instead of computing $\text{CRS}'_\beta[1] = H'(ovk^*_\beta, vid)$, we program the random oracle $H'(\cdot,\cdot)$ so that, on input $(ovk^*_\beta, \cdot)$, instead of returning $\text{CRS}'_\beta[1]$ such that $(\text{CRS}'_\beta[1], \text{CRS}'[2])$ forms a uniform quadruple in $\mathbb{G}_2$, it outputs $\text{CRS}'_\beta[1]$ such that $\text{CRS}'_\beta[1] = \text{CRS}'[2]^{\zeta_\beta}$ for a random scalar $\zeta_\beta \leftarrow \mathbb{Z}_q$ (Assumption 4). This turns CRS'_β, associated to ovk^*_β used in the challenge submission, into simulatable mode for proving multi-scalar equations, and the GS proof π_C is perfectly zero-knowledge (Assumption 2). By the fact that a new *ovk* is generated for each new submission, the adversary A is not able to predict the value of ovk^*_β and thus cannot guess the value of $\text{CRS}'_\beta[1]$. By the indistinguishability property of the GS CRS modes, and the simulation property of GS proofs, the output of Hybrid 2 is indistinguishable from Hybrid 1 with security loss ϵ_{SXDH}.
- Hybrid 3 is identical to Hybrid 2 except that we program the oracle $H(\cdot)$ so that it outputs $H(vid^*)$ on the i^*-th query, for a random $i* \leftarrow \{1, \cdots, Q\}$ and for Q denoting the maximum number of queries made by A to $H(\cdot)$. The oracle outputs $H(vid^*)$, in such a way that $(g, g^{sid^*}, H(vid^*), Z)$ is a random quadruple and we define $C^* = Z$ in the challenge submission. If the i^*-th query is not vid^*, we abort. The probability that the i^*-th query is vid^* is $\frac{1}{Q}$. The output of Hybrid 3 is thus indistinguishable from Hybrid 2 with probability $Q\cdot\epsilon_{SXDH}$. Thanks to the random-self reducibility of SXDH, we do that for both users id_0 and id_1 at the same time. (We refer the reader to the proof of authenticity to see this technique used in a more complex case.)

In Hybrid 3, $\text{EXEC}^b(n, A)$ is independent of the bit b. Indeed, π_σ is perfectly witness-indistinguishable. The proof π_C are simulated for users id_0 and id_1 with independent ovk_1 and ovk_2. The change made in Hybrid 3 makes the tokens C_0 and C_1 uniform and independent

of the view. Therefore, $\text{EXEC}^0(n, A)$ and $\text{EXEC}^1(n, A)$ are exactly the same in Hybrid 3. Hence, the unlinkability.

Security Against Malicious Users. Due to space limitations, the proof for the authenticity of CAnonize is given in Appendix C. The principle is to simulate the submission of honest users while making any adversarially computed proof perfectly sound and extractable. Then, we can show that the adversary necessarily manages to forge one of the signature schemes.

□

6 Performance evaluation

This section presents a detailed evaluation of CAnonize and compares it with the original Anonize scheme [35]. We focus on three complementary aspects: (i) underlying cryptographic assumptions (6.1), (ii) communication cost (6.2), and (iii) computational cost (6.3).

Implementation Details. We built a prototype implementation of both schemes in Rust (v1.85.0) using the `arkworks`(v0.5.0) library [6] for pairing-based cryptography. We use SHA-256 [46] as hash function, and hash-to-curve operations are implemented following RFC9380 [27] using SHA-256 in the hash-to-field function and the map-to-curve function proposed by Wahby and Boneh [56].

To ensure a fair comparison, we used the same libraries and optimizations for both implementations, and no batching techniques were employed that could give an advantage to CAnonize. The prototype implementation is available at https://github.com/uclcrypto/CAnonize.

All experiments were executed 50 times on a machine equipped with an Intel Core i7-1355U CPU (base frequency 1 GHz, turbo frequency up to 5 GHz), 16 GB RAM, running Ubuntu 22.04.1. We report median execution times.

To target a 128-bit security level, both schemes were instantiated over the BLS12-381 curve [11] and the Fischlin transform parameters were fixed to $\lambda = 32$, $b = 4$, and $t = 9$.

6.1 Assumptions

The original Anonize construction relies on three independent cryptographic assumptions: the hardness of the discrete logarithm problem, the decisional bilinear Diffie–Hellman (DBDH) assumption, and the n-Decisional Diffie–Hellman Inversion (n-DDHI) assumption.

In contrast, CAnonize relies solely on the SXDH assumption in the random oracle model and common reference string model. SXDH is a standard assumption in pairing-based cryptography and is widely used in efficient proof systems.

While we do not claim that SXDH is strictly weaker than the combination of assumptions used in Anonize, CAnonize reduces assumption complexity by relying on a single well-studied decisional assumption. This simplification streamlines the security analysis and avoids combining multiple independent hardness assumptions.

6.2 Communication Cost

Table 2 summarizes the communication cost of both schemes. We report (i) the number of transmitted group elements per phase, (ii) the corresponding theoretical size in bits based on curve parameters, and (iii) the experimentally observed serialized size.

Table 2: Communication cost in terms of the number of group elements for our version (CAnonize) and the original version (Anonize), theoretical and experimental communication size in bits

		$\mathbb{Z}_q$	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	theo. size	exp. size
User Reg.	ours		8	3		**5.3k**	**8.5k**
	orig.	96	36	1		39k	62k
Survey Reg.	ours		5	1		2.7k	3.5k
	orig.		2	1		**1.5k**	**2.7k**
Submission	ours	2	38	22		**32k**	**53k**
	orig.	96	64	2	97	494k	564k
Total	ours	2	51	26		**40k**	**65k**
	orig.	192	102	4	97	534k	628k

For BLS12-381, elements have the following theoretical sizes: an element in $\mathbb{Z}_q$ requires 255 bits, an element in $\mathbb{G}_1$ 381 bits, an element in $\mathbb{G}_2$ 762 bits, and an element in $\mathbb{G}_T$ 4572 bits.

Summing all protocol phases, CAnonize transmits a total of 39 753 bits in theory, compared to 534 354 bits for Anonize, corresponding to a 92% reduction in theoretical communication cost. The experimentally measured serialized sizes are 64 512 bits and 628 416 bits, respectively, yielding an empirical reduction of approximately 90%, which closely matches the theoretical reduction.

We note that the communication cost of the survey registration phase is slightly higher in our construction; however, this phase contributes only marginally to the overall communication footprint. The largest savings occur during the submission phase. In particular, CAnonize eliminates the need to transmit any $\mathbb{G}_T$ elements, which are significantly larger than elements in $\mathbb{G}_1$ or $\mathbb{G}_2$. This reduction originates from structural modifications applied to the token generation process and the proof system used during submission. As a result, the communication burden on users is substantially reduced, which is particularly relevant in bandwidth-constrained or large-scale deployments.

6.3 Computational Cost

Table 3 reports the number of cryptographic operations performed by each party in every protocol phase. We count hash evaluations, hash-to-curve operations in $\mathbb{G}_1$ and $\mathbb{G}_2$, pairing computations, multiplications in $\mathbb{G}_T$, and exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_2$. Operations in $\mathbb{Z}_q$ and multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$ are omitted, as they are at least one order of magnitude cheaper than pairing evaluations and exponentiations in $\mathbb{G}_2$.

While the setup, user registration, survey registration, and authorization check phases are slightly more expensive in our construction, the submission phase — which is executed once per survey response — is significantly more efficient. Most notably, CAnonize removes all pairing computations on the user side during submission and substantially reduces their number on the SA side.

Table 3: Number of hashing, hash-to-curve in $\mathbb{G}_1$ and $\mathbb{G}_2$, pairing evaluations, multiplications in $\mathbb{G}_T$, exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_2$ the user, RA and SA in our version (CAnonize) and the original paper (Anonize) for each step of the survey

		Hash.	Hash. curve	Pairing	Mult. $\mathbb{G}_T$	Exp. $\mathbb{G}_1$	Exp. $\mathbb{G}_2$
Setup	RA					1	8
	SA					1	8
	Orig. RA			1		1	
	Orig. SA			1		1	
User Reg.	User			10	6	2	
	RA			8		9	1
	Orig. user	512		2	1	68	
	Orig. RA	32				100	1
Survey Reg.	SA					9	1
	Orig. SA					5	1
Authorized	User			10	6		
	Orig. user			2		2	
Submission	User	1	$1\mathbb{G}_1$, $2\mathbb{G}_2$			83	73
	SA	1	$2\mathbb{G}_2$	101	77	3	
	Orig. user	512		129	64	1155	2
	Orig. SA	32		288	224	288	

To quantify the overall impact, we consider a representative execution flow in which a user registers, is added to a survey, verifies authorization, and submits a single response. Using the per-operation benchmarks reported in Table 4 (hash: 0.57 μs; hash-to-curve $\mathbb{G}_1$: 142 μs; hash-to-curve $\mathbb{G}_2$: 689 μs; pairing: 884 μs; $\mathbb{G}_T$ multiplication: 2 μs; exponentiation in $\mathbb{G}_1$: 4 μs; exponentiation in $\mathbb{G}_2$: 7 μs), the total theoretical execution time is 117 ms for CAnonize and 388 ms for Anonize, corresponding to a 70% reduction in estimated cryptographic cost. Table 4 additionally reports measured median execution times over 50 runs. The total observed execution time is 179 ms for our scheme and 440 ms for Anonize, corresponding to a 59% reduction in wall-clock time. The difference between theoretical and measured times is expected, as the theoretical estimate accounts only for dominant cryptographic operations and does not capture serialization, memory access, data structure handling, or other implementation overheads.

The submission phase accounts for the majority of the improvement: it is 273 ms faster in CAnonize, corresponding to a 67% reduction for that phase alone. The only phases that incur a slight overhead are RA, SA, and CRS setup, user registration, and authorization check, which together increase execution time by approximately 18 ms. These steps are performed infrequently: RA, SA, and CRS setup occurs once, and registration only when users join, whereas submission is executed for each survey response.

Table 4: Median execution time of the different steps of CAnonize and Anonize in ms over 50 runs along with the theoretical total execution time (considering hash: 0.57μs, hash-to-curve $\mathbb{G}_1$: 142μs, hash-to-curve $\mathbb{G}_2$: 689μs, pairing: 884μs, mult $\mathbb{G}_T$: 2μs, exp $\mathbb{G}_1$: 4μs, exp $\mathbb{G}_2$: 7μs)

	Setup			User Reg.		Survey Reg.	
	RA	SA	CRS	User	RA	SA	Auth.
ours	3.0	3.0	6.0	10.0	**6.0**	1.0	8.0
orig.	**1.0**	**1.0**	**0.0**	10.0	8.0	**0.0**	**3.0**

	Submission		Total			Total	
	User	SA	User	RA	SA	Exp.	Theo.
ours	**37.0**	**97.0**	**50.0**	10.0	**102**	**179.0**	**117**
orig.	157.0	250.0	170.5	10.0	254	440.0	388

Overall, the results demonstrate that the proposed modifications yield a substantial reduction in both communication and computational overhead, particularly for the most frequently executed operations, while preserving the security and unlinkability guarantees of the original construction.

7 Conclusion

We presented CAnonize, a new anonymous survey protocol that revisits and improves the design of Anonize [35]. Our objective was to overcome performance bottlenecks in bandwidth and in time, while keeping the original security model and increasing the security guarantees by relying on fewer and better understood computational assumptions.

Our evaluation of CAnonize shows considerable improvements in both communication and computational costs: our prototype implementation achieves a 90% reduction in empirical communication size and a 59% reduction in end-to-end execution time for a representative execution flow. These gains arise from structural modifications to the protocol rather than parameter tuning or implementation artifacts.

From a security perspective, the proposed construction relies solely on the SXDH assumption in the random oracle and common reference string models, replacing the combination of computational assumptions required by Anonize with a single standard decisional assumption.

While CAnonize assumes a trusted common reference string and does not address network-layer metadata leakage, it provides strong cryptographic guarantees: survey responses remain unlinkable even if transcripts or stored data are later exposed.

Overall, our results demonstrate that anonymous survey systems can be significantly streamlined without introducing additional assumptions or weakening security, improving their practicality for large-scale and bandwidth-constrained deployments.

Acknowledgments

The authors used generative AI-based tools to revise the text, improve flow, and correct typos, grammatical errors, and awkward phrasing.

Thomas Peters is a Research Associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS) and partially conducted this work under its Incentive Grant for Scientific Research (MIS). This work has been funded in part by the Walloon Region through the project CyberExcellence (convention number 2110186).

We thank the anonymous reviewers for their insightful comments and suggestions, which have helped us improve the quality of this paper.

References

[1] 2016. Introducing Brave Payments. https://brave.com/blog/introducing-brave-payments/.
[2] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. 2016. Structure-Preserving Signatures and Commitments to Group Elements. *Journal of Cryptology* 29, 2 (April 2016), 363–421. https://doi.org/10.1007/s00145-014-9196-7
[3] Ben Adida. 2008. Helios: Web-based Open-Audit Voting.. In *USENIX security symposium*, Vol. 17. 335–348.
[4] Faris A. Almalki and Ben Othman Soufiene. 2021. EPPDA: An Efficient and Privacy-Preserving Data Aggregation Scheme with Authentication and Authorization for IoT-Based Healthcare Applications. *Wireless Communications and Mobile Computing* 2021, 1 (2021), 5594159. https://doi.org/10.1155/2021/5594159
[5] J Andrew, R Jennifer Eunice, and J Karthikeyan. 2023. An anonymization-based privacy-preserving data collection protocol for digital health data. *Frontiers in public health* 11 (2023), 1125011.
[6] arkworks contributors. 2022. `arkworks` *zkSNARK ecosystem.* https://arkworks.rs
[7] Man Ho Au, Joseph K. Liu, Willy Susilo, and Tsz Hon Yuen. 2006. Constant-Size ID-Based Linkable and Revocable-iff-Linked Ring Signature. In *Progress in Cryptology - INDOCRYPT 2006*, Rana Barua and Tanja Lange (Eds.). Springer, Berlin, Heidelberg, 364–378. https://doi.org/10.1007/11941378_26
[8] Man Ho Au, Joseph K. Liu, Willy Susilo, and Tsz Hon Yuen. 2013. Secure ID-based Linkable and Revocable-Iff-Linked Ring Signature with Constant-Size Construction. *Theoretical Computer Science* 469 (Jan. 2013), 1–14. https://doi.org/10.1016/j.tcs.2012.10.031
[9] Man Ho Au, Joseph K. Liu, Willy Susilo, and Tsz Hon Yuen. 2013. Secure ID-based Linkable and Revocable-Iff-Linked Ring Signature with Constant-Size Construction. *Theoretical Computer Science* 469 (Jan. 2013), 1–14. https://doi.org/10.1016/j.tcs.2012.10.031
[10] Foteini Baldimtsi and Anna Lysyanskaya. 2013. Anonymous Credentials Light. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13)*. Association for Computing Machinery, New York, NY, USA, 1087–1098. https://doi.org/10.1145/2508859.2516687
[11] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. 2003. Constructing Elliptic Curves with Prescribed Embedding Degrees. In *Security in Communication Networks*, Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 257–267.
[12] Adam Bender, Jonathan Katz, and Ruggero Morselli. 2006. Ring Signatures: Stronger Definitions, and Constructions Without Random Oracles. In *Theory of Cryptography*, Shai Halevi and Tal Rabin (Eds.). Springer, Berlin, Heidelberg, 60–79. https://doi.org/10.1007/11681878_4
[13] David Bernhard, Marc Fischlin, and Bogdan Warinschi. 2016. On the Hardness of Proving CCA-Security of Signed ElGamal. In *Public-Key Cryptography - PKC 2016 (Lecture Notes in Computer Science, Vol. 9614)*. Springer, 47–69.
[14] Dan Boneh, Xavier Boyen, and Hovav Shacham. 2004. Short Group Signatures. In *Advances in Cryptology – CRYPTO 2004*, Matt Franklin (Ed.). Springer, Berlin, Heidelberg, 41–55. https://doi.org/10.1007/978-3-540-28628-8_3
[15] Jan Camenisch and Anna Lysyanskaya. 2001. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *Advances in Cryptology — EUROCRYPT 2001*, Birgit Pfitzmann (Ed.). Springer, Berlin, Heidelberg, 93–118. https://doi.org/10.1007/3-540-44987-6_7
[16] Jan Camenisch and Anna Lysyanskaya. 2002. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Advances in Cryptology — CRYPTO 2002*, Moti Yung (Ed.). Springer, Berlin, Heidelberg, 61–76. https://doi.org/10.1007/3-540-45708-9_5
[17] Jan Camenisch and Els Van Herreweghen. 2002. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*. 21–30.
[18] Sébastien Canard, Berry Schoenmakers, Martijn Stam, and Jacques Traoré. 2006. List signature schemes. *Discret. Appl. Math.* 154, 2 (2006), 189–201.
[19] David Chaum, Amos Fiat, and Moni Naor. 1990. Untraceable Electronic Cash. In *Advances in Cryptology — CRYPTO' 88*, Shafi Goldwasser (Ed.). Springer, New York, NY, 319–327. https://doi.org/10.1007/0-387-34799-2_25
[20] David Chaum and Eugène Van Heyst. 1991. Group signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 257–265.
[21] Yi-Hsiu Chen and Yehuda Lindell. 2024. Optimizing and Implementing Fischlin's Transform for UC-Secure Zero Knowledge. *IACR Communications in Cryptology* (July 2024), cc1–2–30. https://doi.org/10.62056/a66chey6b
[22] Sherman SM Chow, Joseph K Liu, and Duncan S Wong. 2008. Robust Receipt-Free Election System with Ballot Secrecy and Verifiability.. In *NDSS*, Vol. 8. 81–94.
[23] Henry Corrigan-Gibbs and Dan Boneh. 2017. Prio: Private, Robust, and Scalable Computation of Aggregate Statistics. In *14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017*. USENIX Association, 259–282.
[24] Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. 2018. Privacy Pass: Bypassing Internet Challenges Anonymously. *Proceedings on Privacy Enhancing Technologies* (2018). https://petsymposium.org/popets/2018/popets-2018-0026.php
[25] Alex Davidson, Peter Snyder, E. B. Quirk, Joseph Genereux, Benjamin Livshits, and Hamed Haddadi. 2022. STAR: Secret Sharing for Private Threshold Aggregation Reporting. In *Proceedings of ACM CCS 2022*. ACM, 697–710.
[26] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. 2004. Anonymous Identification in Ad Hoc Groups. In *Advances in Cryptology - EUROCRYPT 2004*, Christian Cachin and Jan L. Camenisch (Eds.). Springer, Berlin, Heidelberg, 609–626. https://doi.org/10.1007/978-3-540-24676-3_36
[27] A. Faz-Hernandez, S. Scott, N. Sullivan, R. S. Wahby, and C. A. Wood. 2023. *Hashing to Elliptic Curves.* Technical Report RFC9380. RFC Editor. RFC9380 pages. https://doi.org/10.17487/RFC9380
[28] Marc Fischlin. 2005. Communication-Efficient Non-interactive Proofs of Knowledge with Online Extractors. In *Advances in Cryptology - CRYPTO 2005 (Lecture Notes in Computer Science, Vol. 3621)*. Springer, 152–168.
[29] Matthew Franklin and Haibin Zhang. 2013. Unique Ring Signatures: A Practical Construction. In *Financial Cryptography and Data Security*, Ahmad-Reza Sadeghi (Ed.). Springer, Berlin, Heidelberg, 162–170. https://doi.org/10.1007/978-3-642-39884-1_13
[30] Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. 2020. Private Aggregation from Fewer Anonymous Messages. In *Advances in Cryptology – EUROCRYPT 2020*, Anne Canteaut and Yuval Ishai (Eds.). Springer International Publishing, Cham, 798–827. https://doi.org/10.1007/978-3-030-45724-2_27
[31] Jens Groth. 2006. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *Advances in Cryptology – ASIACRYPT 2006*, Xuejia Lai and Kefei Chen (Eds.). Springer, Berlin, Heidelberg, 444–459. https://doi.org/10.1007/11935230_29
[32] Jens Groth and Amit Sahai. 2008. Efficient Non-Interactive Proof Systems for Bilinear Groups. In *Advances in Cryptology – EUROCRYPT 2008*, Nigel Smart (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 415–432.
[33] Meng Hao, Hongwei Li, Guowen Xu, Sen Liu, and Haomiao Yang. 2019. Towards Efficient and Privacy-Preserving Federated Deep Learning. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 1–6. https://doi.org/10.1109/ICC.2019.8761267
[34] Ryan Henry and Ian Goldberg. 2011. Formalizing Anonymous Blacklisting Systems. In *2011 IEEE Symposium on Security and Privacy* (2011-05). IEEE, 81–95. https://doi.org/10.1109/SP.2011.13
[35] Susan Hohenberger, Steven Myers, Rafael Pass, and abhi shelat. 2014. ANONIZE: A Large-Scale Anonymous Survey System. In *2014 IEEE Symposium on Security and Privacy*, Vol. 13. IEEE, 375–389. https://doi.org/10.1109/SP.2014.31
[36] Bjørn Aslak Juliussen, Elisavet Kozyri, Dag Johansen, and Jon Petter Rui. 2023. The third country problem under the GDPR: enhancing protection of data transfers with technology. *International Data Privacy Law* 13, 3 (2023), 225–243.
[37] Charanjit S. Jutla and Arnab Roy. 2017. Improved Structure Preserving Signatures under Standard Bilinear Assumptions. In *Public-Key Cryptography – PKC 2017*, Serge Fehr (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 183–209.
[38] Eungoo Kang and Hee-Joong Hwang. 2023. The Importance of Anonymity and Confidentiality for Conducting Survey Research. *Journal of Research and Publication Ethics* 4, 1 (2023), 1–7. https://doi.org/10.15722/jrpe.4.1.202303.1
[39] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. 2015. End-to-End Verifiable Elections in the Standard Model. In *Advances in Cryptology - EUROCRYPT 2015*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer, Berlin, Heidelberg, 468–498. https://doi.org/10.1007/978-3-662-46803-6_16
[40] Yashvanth Kondi and Abhi Shelat. 2022. Improved Straight-Line Extraction in the Random Oracle Model with Applications to Signature Aggregation. In *Advances in Cryptology - ASIACRYPT 2022 (Lecture Notes in Computer Science, Vol. 13792)*. Springer, 279–309.
[41] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. 2004. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. In *Information Security and Privacy*, Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan (Eds.). Springer, Berlin, Heidelberg, 325–335. https://doi.org/10.1007/978-3-540-27800-9_28
[42] Joseph K. Liu and Duncan S. Wong. 2005. Linkable Ring Signatures: Security Models and New Schemes. In *Computational Science and Its Applications – ICCSA 2005*, Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan (Eds.). Springer, Berlin, Heidelberg, 614–623. https://doi.org/10.1007/11424826_65
[43] Jack P. K. Ma and Sherman S. M. Chow. 2023. SMART Credentials in the Multi-queue of Slackness (or Secure Management of Anonymous Reputation Traits

without Global Halting). In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. 896–912. https://doi.org/10.1109/EuroSP57164.2023.00057

[44] Gonzalo Munilla Garrido, Johannes Sedlmeir, and Matthias Babel. 2022. Towards Verifiable Differentially-Private Polling. In *Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES '22)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3538969.3538992

[45] Moni Naor, Benny Pinkas, and Omer Reingold. 1999. Distributed Pseudo-random Functions and KDCs. In *Advances in Cryptology - EUROCRYPT '99 (Lecture Notes in Computer Science, Vol. 1592)*. Springer, 327–346.

[46] National Institute of Standards and Technology (US). 2015. *Secure Hash Standard*. Technical Report NIST FIPS 180-4. National Institute of Standards and Technology (U.S.), Washington, D.C. NIST FIPS 180–4 pages. https://doi.org/10.6028/NIST.FIPS.180-4

[47] Anthony D. Ong and David J. Weiss. 2000. The Impact of Anonymity on Responses to Sensitive Questions. *Journal of Applied Social Psychology* 30, 8 (2000), 1691–1708. https://doi.org/10.1111/j.1559-1816.2000.tb02462.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1559-1816.2000.tb02462.x

[48] Rafael Pass. 2003. On Deniability in the Common Reference String and Random Oracle Model. In *Advances in Cryptology - CRYPTO 2003 (Lecture Notes in Computer Science, Vol. 2729)*, Dan Boneh (Ed.). Springer, 316–337.

[49] Leonid Reyzin and Sophia Yakoubov. 2016. Efficient Asynchronous Accumulators for Distributed PKI. In *Security and Cryptography for Networks*, Vassilis Zikas and Roberto De Prisco (Eds.). Springer International Publishing, Cham, 292–309. https://doi.org/10.1007/978-3-319-44618-9_16

[50] Ronald L Rivest, Adi Shamir, and Yael Tauman. 2001. How to leak a secret. In *International conference on the theory and application of cryptology and information security*. Springer, 552–565.

[51] Ronald L. Rivest, Adi Shamir, and Yael Tauman. 2006. How to Leak a Secret: Theory and Applications of Ring Signatures. In *Theoretical Computer Science: Essays in Memory of Shimon Even*, Oded Goldreich, Arnold L. Rosenberg, and Alan L. Selman (Eds.). Springer, Berlin, Heidelberg, 164–186. https://doi.org/10.1007/11685654_7

[52] Michael Rosenberg, Mary Maller, and Ian Miers. 2022. SNARKBlock: Federated Anonymous Blocklisting from Hidden Common Input Aggregate Proofs. In *2022 IEEE Symposium on Security and Privacy (SP)*. 948–965. https://doi.org/10.1109/SP46214.2022.9833656

[53] Victor Shoup and Rosario Gennaro. 2002. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. *J. Cryptol.* 15, 2 (2002), 75–96.

[54] Amit Singh. 2025. From past to present: the evolution of data breach causes (2005–2025). *LatIA* 3 (2025), 333–333.

[55] Nam Tran, Khoa Nguyen, Dongxi Liu, Josef Pieprzyk, and Willy Susilo. 2026. Many-Time Linkable Ring Signatures. In *Provable and Practical Security*, Guomin Yang, Shengli Liu, Chunhua Su, Akira Otsuka, and Zhuotao Lian (Eds.). Springer Nature, Singapore, 3–22. https://doi.org/10.1007/978-981-95-2961-2_1

[56] Riad S. Wahby and Dan Boneh. 2019. Fast and Simple Constant-Time Hashing to the BLS12-381 Elliptic Curve. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (Aug. 2019), 154–179. https://doi.org/10.13154/tches.v2019.i4.154-179

[57] Joshua M Wilson. 2022. Cross-Border Data Transfers: A Balancing Act through Federal Law. *Bus. Entrepreneurship & Tax L. Rev.* 6 (2022), 150.

[58] Min Xie, Zhengzhou Tu, Man Ho Au, Junbin Fang, Xuan Wang, and Zoe Lin Jiang. 2025. Efficient Constant-Size Linkable Ring Signatures for Ad-Hoc Rings via Pairing-Based Set Membership Arguments. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25)*. Association for Computing Machinery, New York, NY, USA, 858–872. https://doi.org/10.1145/3719027.3744830

[59] Xin Xie and Yu-Chi Chen. 2021. Decentralized Data Aggregation: A New Secure Framework Based on Lightweight Cryptographic Algorithms. *Wireless Communications and Mobile Computing* 2021, 1 (2021), 5565663. https://doi.org/10.1155/2021/5565663

[60] Tao Zhang, Huangting Wu, and Sherman S. M. Chow. 2019. Structure-Preserving Certificateless Encryption and Its Application. In *Topics in Cryptology – CT-RSA 2019*, Mitsuru Matsui (Ed.). Springer International Publishing, Cham, 1–22. https://doi.org/10.1007/978-3-030-12612-4_1

[61] Mingxun Zhou, Giulia Fanti, and Elaine Shi. 2024. Conan: Distributed Proofs of Compliance for Anonymous Data Collection. In *Proceedings of ACMCCS 2024*. ACM, 914–928.

A Groth–Sahai proofs

Table 5: Commitments in Groth–Sahai proofs

Committed value	Commitment
$X \in \mathbb{G}_1$	$c_X = \begin{pmatrix} g_{11}^{r_1} g_{21}^{r_2} \\ X g_{12}^{r_1} g_{21}^{r_2} \end{pmatrix}$ $\quad r_1, r_2 \leftarrow \mathbb{Z}_q$
$\hat{Y} \in \mathbb{G}_2$	$\hat{c}_{\hat{Y}} = \begin{pmatrix} \hat{g}_{11}^{s_1} \hat{g}_{21}^{s_2} \\ \hat{Y} \hat{g}_{12}^{s_1} \hat{g}_{21}^{s_2} \end{pmatrix}$ $\quad s_1, s_2 \leftarrow \mathbb{Z}_q$
$y \in \mathbb{Z}_q$	$c_y = \begin{pmatrix} \hat{g}_{21}^{y} \hat{g}_{11}^{s} \\ \hat{g}_{22}^{y} \hat{g}_{11}^{y} \hat{g}_{12}^{s} \end{pmatrix}$ $\quad s \leftarrow \mathbb{Z}_q$

Table 6: Proofs for pairing product and multi-scalar multiplication equations

Equation	Proof
$e(X, \hat{B}) = T$	$\hat{\pi}_X = \begin{pmatrix} \hat{B}^{r_1} \\ \hat{B}^{r_2} \end{pmatrix}$
$e(X, \hat{Y}) = T$	$\hat{\pi}_Y = \left[\begin{pmatrix} 1 \\ \hat{Y} \end{pmatrix}^{r_1} \vec{\hat{g}}_1^{\,s_1 \cdot r_1 - t_1} \vec{\hat{g}}_2^{\,s_2 \cdot r_1 - t_3}, \begin{pmatrix} 1 \\ \hat{Y} \end{pmatrix}^{r_2} \vec{\hat{g}}_1^{\,s_1 \cdot r_2 - t_2} \vec{\hat{g}}_2^{\,s_2 \cdot r_2 - t_4} \right]$ $\quad t_1, t_2, t_3, t_4 \leftarrow \mathbb{Z}_q$ $\theta_X = \left[\begin{pmatrix} 1 \\ X \end{pmatrix}^{s_1} \vec{g}_1^{\,t_1} \vec{g}_2^{\,t_2}, \begin{pmatrix} 1 \\ X \end{pmatrix}^{s_2} \vec{g}_1^{\,t_3} \vec{g}_2^{\,t_4} \right]$
$A^y = T$	$\theta_y = \begin{pmatrix} 1 \\ A \end{pmatrix}^{s}$

Table 7: Verification equations for pairing product and multi-scalar multiplication equations

Equation	Verification Equation
$e(X, \hat{B}) = T$	$E(c_X, \begin{pmatrix} 1 \\ \hat{B} \end{pmatrix}) \stackrel{?}{=}$ $\begin{pmatrix} 1 & 1 \\ 1 & T \end{pmatrix} \cdot E(\vec{g}_1, \hat{\pi}_X[1]) \cdot E(\vec{g}_2, \hat{\pi}_X[2])$
$e(X, \hat{Y}) = T$	$E\left(\begin{pmatrix} 1 \\ A \end{pmatrix}, \hat{c}_Y \right) \cdot E\left(c_X, \begin{pmatrix} 1 \\ B \end{pmatrix} \right) \stackrel{?}{=}$ $\begin{pmatrix} 1 & 1 \\ 1 & T \end{pmatrix} \cdot E(\vec{g}_1, \hat{\pi}_X[1]) \cdot E(\vec{g}_2, \hat{\pi}_X[2]) \cdot$ $E(\theta_Y[1], \vec{\hat{g}}_1) \cdot E(\theta_Y[2], \vec{\hat{g}}_2)$
$A^y = T$	$E(c_X, c_y) \stackrel{?}{=}$ $\begin{pmatrix} 1 & 1 \\ e(T, \hat{g}_{21}) & e(T, \hat{g}_{22}\hat{g}_{11}) \end{pmatrix} \cdot E(\theta_y, \vec{\hat{g}}_1)$

B Detailed proofs of knowledge using Groth–Sahai

B.1 Notation

In the following, we use the following notation for denoting each element of the different CRS:

$$\mathrm{CRS}_1 = \begin{pmatrix} g_{11} & g_{21} \\ g_{12} & g_{22} \end{pmatrix} \mathrm{CRS}_2 = \begin{pmatrix} \hat{g}_{11} & \hat{g}_{21} \\ \hat{g}_{12} & \hat{g}_{22} \end{pmatrix} \mathrm{CRS}' = \begin{pmatrix} \hat{g}'_{11} & \hat{g}'_{21} \\ \hat{g}'_{12} & \hat{g}'_{22} \end{pmatrix}$$

We also use the following notation for denoting each vector of each CRS:

$$\vec{g}_1 = \begin{pmatrix} g_{11} \\ g_{12} \end{pmatrix}, \quad \vec{g}_2 = \begin{pmatrix} g_{21} \\ g_{22} \end{pmatrix} \quad , \cdots \quad \text{and } \mathrm{CRS}'[1] = \widehat{g}_1, \mathrm{CRS}'[2] = \widehat{g}_2$$

We denote pairing operations as follows:

$$E_1(x, \vec{y}) = \begin{pmatrix} e(x, y_1) \\ e(x, y_2) \end{pmatrix}, x \in \mathbb{G}_1, \vec{y} \in \mathbb{G}_2^2$$

$$E_2(\vec{x}, y) = \begin{pmatrix} e(x_1, y) \\ e(x_2, y), \end{pmatrix}, \vec{x} \in \mathbb{G}_1^2, y \in \mathbb{G}_2$$

$$E_T(\vec{x}, \vec{y}) = \begin{pmatrix} e(x_1, y_1) & e(x_2, y_1) \\ e(x_1, y_2) & e(x_2, y_2) \end{pmatrix}, \vec{x} \in \mathbb{G}_1^2, \vec{y} \in \mathbb{G}_2^2$$

B.2 Knowledge of sid in user registration

We want to prove:

$$PoK\{(sid) : pk = g^{sid}\}$$

For this equation, the commitment is:

$$c_{sid} = \begin{pmatrix} 1 \\ \hat{g}_{11}^{sid} \end{pmatrix} \vec{\hat{g}}_1^{\,r^{sid}} \vec{\hat{g}}_2^{\,sid}$$

The proof element is:

$$\theta_{sid} = g^{r^{sid}}$$

The verification equation is :

$$E_1(g, \hat{c}_{sid}) \stackrel{?}{=} E_2\left(pk, \begin{pmatrix} 1 \\ \hat{g}_{11} \end{pmatrix} \hat{g}_2\right) \cdot E_2(\theta_{sid}, \hat{g}_1) \qquad \text{(UR)}$$

B.2.1 GenProof$_{\mathrm{UR}}$ and VerifyProof$_{\mathrm{UR}}$. The proof generation algorithm GenProof$_{\mathrm{UR}}$ and the proof verification algorithm VerifyProof$_{\mathrm{UR}}$ are defined as follows:

$$\mathrm{GenProof}_{\mathrm{UR}}(\mathrm{CRS}_2) \rightarrow \pi_{\mathrm{UR}} : (c_{sid}, \theta_{sid})$$

$$\mathrm{VerifyProof}_{\mathrm{UR}}(\mathrm{CRS}_2, \pi_{\mathrm{UR}}) \rightarrow \{0, 1\} :$$
Output 1 if equation (UR) holds,
else output 0.

B.3 Satisfaction of condition (a)

We want to prove the knowledge of the witnesses that satisfy the verification equations of the SPS signature from the RA on (g^{id}, pk) and the SPS signature from the RA on (g^{id}, vid).

$$\text{(a) } PoK \left\{ \begin{array}{l} (g^{id}, pk, \rho, \tilde{\rho}, \psi, \gamma, \hat{\tau}, \rho', \tilde{\rho}', \psi', \gamma', \hat{\tau}') : \\ \quad e(g^{id}, \hat{C}_{01}) e(pk, \hat{C}_{02}) e(\rho, \hat{C}_{03}) e(\tilde{\rho}, \hat{C}_{04}) \cdot \\ \quad e(\psi, \hat{C}_{05}) e(\gamma, \hat{C}_{06}) e(h, C_1) = e(\pi, C_2) \\ \quad \wedge e(\rho, \hat{\tau}) = e(\psi, \hat{h}) \\ \quad \wedge e(g^{id}, \hat{C}'_{01}) e(g^{vid}, \hat{C}'_{02}) e(\rho', \hat{C}'_{03}) e(\tilde{\rho}', \hat{C}'_{04}) \cdot \\ \quad e(\psi', \hat{C}'_{05}) e(\gamma', \hat{C}'_{06}) e(h', \hat{C}'_1) = e(\pi', \hat{C}'_2) \\ \quad \wedge e(\rho', \hat{\tau}') = e(\psi', \hat{h}') \end{array} \right\}$$

$$\text{with } pk_{RA} = (C_0 = \begin{pmatrix} \hat{C}_{01} = \hat{h}^{k_0 \cdot a} \\ \vdots \\ \hat{C}_{06} = \hat{h}^{k_6 \cdot a} \end{pmatrix}, C_1 = \hat{h}^{k\dot{a}}, C_2 = \hat{h}^{a}).$$

$$\text{and } pk_{SA} = (C'_0 = \begin{pmatrix} \hat{C}'_{01} = \hat{h}'^{k'_0 \cdot a'} \\ \vdots \\ \hat{C}'_{06} = \hat{h}'^{k'_6 \cdot a'} \end{pmatrix}, C'_1 = \hat{h}'^{k' \dot{a}'}, C'_2 = h h'^{a'}).$$

We proceed separately for each equation.

B.3.1 Equation (a.1).

$$\text{(a.1) } PoK \left\{ \begin{array}{l} (g^{id}, pk, \rho, \tilde{\rho}, \psi, \gamma) : \\ \quad e(g^{id}, \hat{h}^{k_1 \cdot a}) e(pk, \hat{h}^{k_2 \cdot a}) e(\rho, \hat{h}^{k_3 \cdot a}) e(\tilde{\rho}, \hat{h}^{k_4 \cdot a}) \cdot \\ \quad e(\psi, \hat{h}^{k_5 \cdot a}) e(\gamma, \hat{h}^{k_6 \cdot a}) e(h, \hat{h}^{a}) = e(\pi, \hat{h}^{a}). \end{array} \right\}$$

For this equation, the commitments are:

$$c_\rho = \begin{pmatrix} 1 \\ \rho \end{pmatrix} \vec{g}_1^{\,r_1^\rho} \vec{g}_2^{\,r_2^\rho} \qquad c_\psi = \begin{pmatrix} 1 \\ \psi \end{pmatrix} \vec{g}_1^{\,r_1^\psi} \vec{g}_2^{\,r_2^\psi} \qquad c_{g^{id}} = \begin{pmatrix} 1 \\ g^{id} \end{pmatrix} \vec{g}_1^{\,r_1^{g^{id}}} \vec{g}_2^{\,r_2^{g^{id}}}$$

$$c_{\tilde{\rho}} = \begin{pmatrix} 1 \\ \tilde{\rho} \end{pmatrix} \vec{g}_1^{\,r_1^{\tilde{\rho}}} \vec{g}_2^{\,r_2^{\tilde{\rho}}} \qquad c_\gamma = \begin{pmatrix} 1 \\ \gamma \end{pmatrix} \vec{g}_1^{\,r_1^\gamma} \vec{g}_2^{\,r_2^\gamma} \qquad c_{pk} = \begin{pmatrix} 1 \\ pk \end{pmatrix} \vec{g}_1^{\,r_1^{pk}} \vec{g}_2^{\,r_2^{pk}}$$

$$c_\pi = \begin{pmatrix} 1 \\ \pi \end{pmatrix} \vec{g}_1^{\,r_1^\pi} \vec{g}_2^{\,r_2^\pi}$$

The proof elements are:

$$\hat{\pi}_{a1,1} = \hat{C}_{01}^{r_1^{g^{id}}} \cdot \hat{C}_{02}^{r_1^{pk}} \cdot \hat{C}_{03}^{r_1^\rho} \cdot \hat{C}_{04}^{r_1^{\tilde{\rho}}} \cdot \hat{C}_{05}^{r_1^\psi} \cdot \hat{C}_{06}^{r_1^\gamma} \cdot (\hat{C}_2^{r_1^\pi})^{-1}$$

$$\hat{\pi}_{a1,2} = \hat{C}_{01}^{r_2^{g^{id}}} \cdot \hat{C}_{02}^{r_2^{pk}} \cdot \hat{C}_{03}^{r_2^\rho} \cdot \hat{C}_{04}^{r_2^{\tilde{\rho}}} \cdot \hat{C}_{05}^{r_2^\psi} \cdot \hat{C}_{06}^{r_2^\gamma} \cdot (\hat{C}_2^{r_2^\pi})^{-1}$$

The verification equation is :

$$E_1(c_\rho, \hat{C}_{03}) E_1(c_{\tilde{\rho}}, \hat{C}_{04}) E_1(c_\psi, \hat{C}_{05}) E_1(c_\gamma, \hat{C}_{06}) \cdot E_1(c_\pi, \hat{C}_1)^{-1} E_1(c_{g^{id}}, \hat{C}_{01}) E_1(c_{pk}, \hat{C}_{02}) \stackrel{?}{=} \begin{pmatrix} 1 \\ T_1 \end{pmatrix} E_1(\vec{g}_1, \pi_{a1,1}) E_1(\vec{g}_2, \pi_{a1,2}) \qquad \text{(RA1)}$$

with $T_1 = e(h, \hat{h}^{k \cdot a})^{-1}$.

B.3.2 Equation (a.2).

$$\text{(a.2) } PoK \left\{ \begin{array}{l} (\rho, \hat{\tau}, \psi) : \\ \quad e(\rho, \hat{\tau}) e(\psi, \hat{h})^{-1} = 1 \end{array} \right\}$$

The commitment is:

$$\hat{c}_{\hat{\tau}} = \begin{pmatrix} 1 \\ \hat{\tau} \end{pmatrix} \vec{\hat{g}}_1^{\,r_1^{\hat{\tau}}} \vec{\hat{g}}_2^{\,r_2^{\hat{\tau}}}$$

The proof elements are:

$$\hat{\pi}_{a2,1} = \begin{pmatrix} 1 \\ \hat{\tau}^{r_1^{rho}} \cdot (\hat{h}^{r_1^{\psi}})^{-1} \end{pmatrix} \vec{\hat{g}}_1^{\,r_1^{\tau} r_1^{\rho} - \nu} \vec{\hat{g}}_2^{\,r_2^{\tau} r_1^{\rho} - \nu^{**}} \qquad \theta_{a2,1} = \begin{pmatrix} 1 \\ \rho^{r_1^{\hat{\tau}}} \end{pmatrix} \vec{g}_1^{\nu} \vec{g}_2^{\nu^*}$$

$$\hat{\pi}_{a2,2} = \begin{pmatrix} 1 \\ \hat{\tau}^{r_2^{\rho}} \cdot (\hat{h}^{r_2^{\psi}})^{-1} \end{pmatrix} \vec{\hat{g}}_1^{\,r_1^{\tau} r_2^{\rho} - \nu^*} \vec{\hat{g}}^{\,r_2^{\tau} r_2^{\rho} - \nu^{***}} \qquad \theta_{a2,2} = \begin{pmatrix} 1 \\ \rho^{r_2^{\hat{\tau}}} \end{pmatrix} \vec{g}_1^{\nu^{**}} \vec{g}_2^{\nu^{***}}$$

with $\nu, \nu^*, \nu^{**}, \nu^{***} \leftarrow \mathbb{Z}_q$. The verification equation is given by:

$$E_T(c_\rho, c_{\hat{\tau}}) E_T(c_\psi, \begin{pmatrix} 1 \\ \hat{h}^{-1} \end{pmatrix}) \stackrel{?}{=}$$
$$E_T(\vec{g}_1, \hat{\pi}_{a2,1}) E_T(\vec{g}_2, \hat{\pi}_{a2,2}) E_T(\theta_{a2,1}, \vec{\hat{g}}_1) E_T(\theta_{a2,2}, \vec{\hat{g}}_2) \qquad \text{(RA2)}$$

B.3.3 Equation (a.3). The satisfaction of SA's verification equations is done similarly to that of the RA's verification equations.

$$\text{(a.3) } PoK \left\{ \begin{array}{l} (g^{id}, \rho', \tilde{\rho}', \psi', \gamma') : \\ e(g^{id}, \hat{h}'^{k'_1 \cdot a'}) e(g^{vid}, \hat{h}'^{k'_2 \cdot a'}) e(\rho', \hat{h}'^{k'_3 \cdot a'}) e(\tilde{\rho}', \hat{h}'^{k'_4 \cdot a'}) \cdot \\ e(\psi', \hat{h}'^{k'_5 \cdot a'}) e(\gamma', \hat{h}'^{k'_6 \cdot a'}) e(h', \hat{h}'^{a'}) = e(\pi', \hat{h}'^{a'}). \end{array} \right\}$$

For this equation, the commitments are:

$$c_{\rho'} = \begin{pmatrix} 1 \\ \rho' \end{pmatrix} \vec{g}_1^{\,r_1^{\rho'}} \vec{g}_2^{\,r_2^{\rho'}} \qquad c_{\psi'} = \begin{pmatrix} 1 \\ \psi' \end{pmatrix} \vec{g}_1^{\,r_1^{\psi'}} \vec{g}_2^{\,r_2^{\psi'}}$$
$$c_{\tilde{\rho}'} = \begin{pmatrix} 1 \\ \tilde{\rho}' \end{pmatrix} \vec{g}_1^{\,r_1^{\tilde{\rho}'}} \vec{g}_2^{\,r_2^{\tilde{\rho}'}} \qquad c_{\gamma'} = \begin{pmatrix} 1 \\ \gamma' \end{pmatrix} \vec{g}_1^{\,r_1^{\gamma'}} \vec{g}_2^{\,r_2^{\gamma'}}$$
$$c_{\pi'} = \begin{pmatrix} 1 \\ \pi' \end{pmatrix} \vec{g}_1^{\,r_1^{\pi'}} \vec{g}_2^{\,r_2^{\pi'}}$$

Note that the commitment $c_{g^{id}}$ is reused from section B.3.1 because we want to prove we use the same value g^{id} in both proofs.

The proof elements are:

$$\hat{\pi}_{a3,1} = \hat{C}_{01}'^{\,r_1^{g^{id}}} \cdot \hat{C}_{03}'^{\,r_1^{\rho'}} \cdot \hat{C}_{04}'^{\,r_1^{\tilde{\rho}'}} \cdot \hat{C}_{05}'^{\,r_1^{\psi'}} \cdot \hat{C}_{06}'^{\,r_1^{\gamma'}} \cdot (\hat{C}_2'^{\,r_1^{\pi'}})^{-1}$$
$$\hat{\pi}_{a3,2} = \hat{C}_{01}'^{\,r_2^{g^{id}}} \cdot \hat{C}_{03}'^{\,r_2^{\rho'}} \cdot \hat{C}_{04}'^{\,r_2^{\tilde{\rho}'}} \cdot \hat{C}_{05}'^{\,r_2^{\psi'}} \cdot \hat{C}_{06}'^{\,r_2^{\gamma'}} \cdot (\hat{C}_2'^{\,r_2^{\pi'}})^{-1}$$

The verification equation is :

$$E_1(c_{\rho'}, \hat{C}'_{03}) E_1(c_{\tilde{\rho}'}, \hat{C}'_{04}) E_1(c_{\psi'}, \hat{C}'_{05}) E_1(c_{\gamma'}, \hat{C}'_{06}) \cdot$$
$$E_1(c_{\pi'}, C'_2)^{-1} E_1(c_{g^{id}}, \hat{C}'_{01}) E_1(c_{pk}, \hat{C}'_{02})$$
$$\stackrel{?}{=} \begin{pmatrix} 1 \\ T'_1 \end{pmatrix} E_1(\vec{g}_1, \hat{\pi}_{a3,1}) E_1(\vec{g}_2, \hat{\pi}_{a3,2}) \qquad \text{(SA1)}$$

with $T'_1 = e(g^{vid}, \hat{C}'_{02})^{-1} e(h', C'_2)^{-1}$.

B.3.4 Equation (a.4).

$$\text{(a.4) } PoK \left\{ \begin{array}{l} (\rho', \hat{\tau}', \psi') : \\ e(\rho', \hat{\tau}') e(\psi', \hat{h}')^{-1} = 1 \end{array} \right\}$$

The commitment is:

$$\hat{c}_{\hat{\tau}'} = \begin{pmatrix} 1 \\ \hat{\tau}' \end{pmatrix} \vec{\hat{g}}_1^{\,r_1^{\hat{\tau}'}} \vec{\hat{g}}_2^{\,r_2^{\hat{\tau}'}}$$

The proof elements are:

$$\hat{\pi}_{a4,1} = \begin{pmatrix} 1 \\ \hat{\tau}' r_1^{\rho'} \cdot (\hat{h}^{r_1^{\psi'}})^{-1} \end{pmatrix} \vec{\hat{g}}_1^{\,r_1^{\tau'} r_1^{\rho'} - \nu'} \vec{\hat{g}}_2^{\,r_2^{\tau'} r_1^{\rho'} - \nu'^{**}}$$

$$\hat{\pi}_{a4,2} = \begin{pmatrix} 1 \\ \hat{\tau}' r_2^{\rho'} \cdot (\hat{h}^{r_2^{\psi'}})^{-1} \end{pmatrix} \vec{\hat{g}}_1^{\,r_1^{\tau'} r_2^{\rho'} - \nu'^{*}} \vec{\hat{g}}_2^{\,r_2^{\tau'} r_2^{\rho'} - \nu'^{***}}$$

$$\theta_{a4,1} = \begin{pmatrix} 1 \\ \rho^{r_1^{\hat{\tau}'}} \end{pmatrix} \vec{g}_1^{\nu'} \vec{g}_2^{\nu'^*}$$

$$\theta_{a4,2} = \begin{pmatrix} 1 \\ \rho^{r_2^{\hat{\tau}'}} \end{pmatrix} \vec{g}_1^{\nu'^{**}} \vec{g}_2^{\nu'^{***}}$$

with $\nu', \nu'^*, \nu'^{**}, \nu'^{***} \leftarrow \mathbb{Z}_q$ The verification equation is given by:

$$E_1(c_{\rho'}, c_{\hat{\tau}'}) E_1(c_{\psi'}, \begin{pmatrix} 1 \\ \hat{h}' \end{pmatrix})^{-1} \stackrel{?}{=}$$
$$E_T(\vec{g}_1, \hat{\pi}_{a4,1}) E_T(\vec{g}_2, \hat{\pi}_{a4,2}) E_T(\theta_{a4,1}, \vec{\hat{g}}_1) E_T(\theta_{a4,2}, \vec{\hat{g}}_2) \qquad \text{(SA2)}$$

B.3.5 GenProof$_\sigma$ and VerifyProof$_\sigma$. The proof generation algorithm GenProof$_\sigma$ and the proof verification algorithm VerifyProof$_\sigma$ are defined as follows:

$$\text{GenProof}_\sigma(\text{CRS}_1, \text{CRS}_2) \rightarrow \pi_\sigma : (c_\rho, c_{\tilde{\rho}}, c_\psi, c_\gamma, c_\pi, c_{g^{id}}, c_{pk}, c_{\hat{\tau}},$$
$$\hat{\pi}_{a1,1}, \hat{\pi}_{a1,2}, \hat{\pi}_{a2,1}, \hat{\pi}_{a2,2}, \theta_{a2,1}, \theta_{a2,2},$$
$$c_{\rho'}, c_{\tilde{\rho}'}, c_{\psi'}, c_{\gamma'}, c_{\pi'}, c_{\hat{\tau}'},$$
$$\hat{\pi}_{a3,1}, \hat{\pi}_{a3,2}, \hat{\pi}_{a4,1}, \hat{\pi}_{a4,2}, \theta_{a4,1}, \theta_{a4,2})$$

$\text{VerifyProof}_\sigma(\text{CRS}_1, \text{CRS}_2, \pi_\sigma) \rightarrow \{0, 1\}$:

Output 1 if equations (RA1), (RA2), (SA1) and (SA2) hold, else output 0.

B.4 Satisfaction of condition (b)

Here we want to prove that the same value of *sid* was used in c_{pk} and in C. More formally, we want to prove:

$$\text{(b) } PoK \left\{ \begin{array}{l} (sid, r_1^{pk}, r_2^{pk}) : \\ \begin{pmatrix} 1 \\ g^{sid} \end{pmatrix} \vec{g}_1^{\,r_1^{pk}} \vec{g}_2^{\,r_2^{pk}} = c_{pk} \\ \wedge\, H(vid)^{sid} = C \end{array} \right\}$$

As we need to prove a statement about a commitment, we use another CRS CRS′. The component CRS′[2] is fixed and CRS′[1] = $H'(ovk, vid)$ = is generated during the submission.

B.4.1 Equation (b.1).

$$\text{(b.1) } PoK \left\{ \begin{array}{l} (sid, r_1^{pk}, r_2^{pk}) : \\ \begin{pmatrix} 1 \\ g^{sid} \end{pmatrix} \vec{g}_1^{\,r_1^{pk}} \vec{g}_2^{\,r_2^{pk}} = c_{pk} \end{array} \right\}$$

The commitments are:

$$\hat{c}_{sid} = \begin{pmatrix} 1 \\ \hat{g}_{11'}^{sid} \end{pmatrix} \vec{g}_2^{\,sid} \vec{g}_1^{\,\rho_{sid}} \qquad \hat{c}_{r_1^{pk}} = \begin{pmatrix} 1 \\ \hat{g}_{11'}^{r_1^{pk}} \end{pmatrix} \vec{g}_2^{\,r_1^{pk}} \vec{g}_1^{\,\rho_{r_1}}$$

$$\hat{c}_{r_2^{pk}} = \begin{pmatrix} 1 \\ \hat{g}_{11'}^{r_2^{pk}} \end{pmatrix} \vec{g}_2^{\,r_2^{pk}} \vec{g}_1^{\,\rho_{r_2}}$$

The proof element is:

$$\theta_{b1} = \begin{pmatrix} g_{11'}^{\rho_{r_1}} g_{21'}^{\rho_{r_2}} \\ g^{\rho_{sid}} g_{12'}^{\rho_{r_1}} g_{22'}^{\rho_{r_2}} \end{pmatrix}$$

The verification equation is:

$$E_T\left(\begin{pmatrix} 1 \\ g \end{pmatrix}, \hat{c}_{sid}\right) E_T(\vec{g}_1, \hat{c}_{r_1}) E_T(\vec{g}_2, \hat{c}_{r_2}) \stackrel{?}{=} E_T(c_{pk}, \begin{pmatrix} 1 \\ \hat{g}'_{11} \end{pmatrix} \vec{g}_2') E(\theta_{b1}, \vec{g}_1') \tag{C1}$$

B.4.2 Equation (b.2).

$$\text{(b.2) } PoK \left\{ \begin{array}{l} (sid): \\ \quad H(vid)^{sid} = C \end{array} \right\}$$

The proof element is:

$$\theta_{b2} = H(vid)^{\rho_{sid}}$$

The verification equation is:

$$E_2(H(vid), \hat{c}_{sid}) \stackrel{?}{=} E_2(C, \begin{pmatrix} 1 \\ \hat{g}'_{11} \end{pmatrix} \vec{g}_2') E_2(\theta_{b2}, \vec{g}_1') \tag{C2}$$

B.4.3 GenProof$_C$ and VerifyProof$_C$. The proof generation algorithm GenProof$_C$ and the proof verification algorithm VerifyProof$_C$ are defined as follows:

$$\text{GenProof}_C(\text{CRS}') \to \pi_C : (\hat{c}_{sid}, \hat{c}_{r_1^{pk}}, \hat{c}_{r_2^{pk}}, \theta_{b1}, \theta_{b2})$$

$$\text{VerifyProof}_C(\text{CRS}', \pi_C) \to \{0,1\}:$$
Output 1 if equations (C1) and (C2) hold,
else output 0.

C Proof of Authenticity

Security against malicious users. Let A be a non-uniform PPT adversary against the authenticity of the anonymous survey. We consider the following sequence of hybrid experiments:

- Hybrid 0 is the real security game given in Definition 10. The public-coin CRSes CRS_{UR} for multi-scalar multiplication and $\text{CRS}_1, \text{CRS}_2$ for pairing product equations are all uniform. This is also the case for all the CRS′ complemented with $\text{CRS}'[1] = H'(ovk, vid)$ in the ROM, and involved in any valid submission.
- In Hybrid 1, the uniformly random $\text{CRS}_{\text{UR}} \in \mathbb{G}_2^4$ is turned into a CRS to allow simulating the proof (of knowledge of $\hat{g}^{sid}$) that pk is well-formed. Under SXDH or Assumption 4, this change is indistinguishable. Let PK_{honest} and $PK_{corrupt}$ be the sets of pk successfully registered by honest and corrupt users, respectively. We stress that the adversary may try to reuse an honestly generated pk when registering a corrupt user with another identity. During an honest registration, the proof associated to $pk \in PK_{honest}$ are simulated, even if we still generate them as $pk = g^{sid}$ for known $sid \leftarrow \mathbb{Z}_q$. If the adversary queries submission for honest users, we still use sid as in Hybrid 0, i.e., as in the genuine run of the Submission algorithm of the scheme.
- In Hybrid 2, $\text{CRS}_1 = (g_{11}, g_{12}, g_{21}, g_{22})$ and CRS_2 for the pairing product equations associated to the verifiability of the structure-preserving signatures are now generated in the binding mode. That is, we no more generated uniform quadruples in $\mathbb{G}_1$ and $\mathbb{G}_2$ but linearly independent couples of vectors. This requires one transition under SXDH for each CRS and the change is indistinguishable (Assumption 4). Now, each valid proof π_σ allows extracting from the commitments group elements that satisfy the verification equations with probability one (Assumption 3). In particular, we extract pk from c_{pk} as well as ID in $\mathbb{G}$ along with valid signatures σ_{RA} on (g^{id}, pk) and $\sigma_{id,vid}$ on (g^{id}, vid).
- In Hybrid 3, the distribution of each CRS′ that has been completed with $\text{CRS}'[1] = H'(ovk, vid)$ for proving correctness of token generation is now turned into the hiding mode for multi-scalar multiplications as CRS_{UR} above. By programming each output of H' and by relying on the random-self reducibility of SXDH, the transition requires a single reduction from SXDH as in Assumption 4. Now, to answer any Submission query, we simulate the proofs π_C that c_{pk} and $C = H(vid)^{sid}$ relies on the same sid related to some $pk \in PK_{honest}$. Nevertheless, we stress that we still use honest sid when computing π_C that c_{pk} is well-formed during a submission, as well as $pk \in PK_{honest}$ in the registration. According to Assumption 2, this simulation is perfect.
- Hybrid 4 is as Hybrid 3, except that we introduce a failure event which causes the challenger to abort the game and outputs `reject` if it occurs. This failure event is defined as follows. Let us denote by ovk_i the one-time verification key generated when answering the i-th Submission query. At that time, the challenger needs to compute $\text{CRS}'_i[1] = H'(ovk_i, vid)$ for the involved survey identifier vid chosen by the adversary. Now, by definition, the failure event occurs if the random oracle has already defined an output for (ovk_i, vid). Since all the ovk_i's of the one-time signature scheme (recalled in Section 3.6) are generated afresh in any Submission query and that it contains two uniformly random group elements of $\mathbb{G}_1$, the probability that the adversary already queried H' on any of these ovk_i's throughout the whole game before it is generated in the i-th Submission query is bounded by $(Q')^2/q^2$, where Q' is the total number of H' queries. Since H' is also queried, but by the challenger, in answering Submission queries, if the failure event does not happen, all the ovk_i's are distinct.
- Hybrid 5 is as Hybrid 4, but we introduce a second failure event so that the challenger also aborts the game and outputs `reject` if it occurs. This failure event is defined as the event that the adversary produces a valid submission $Sub = (C, m, \pi_\sigma, \pi_C, \sigma_O, ovk)$ distinct from any submission

answers $Sub_i = (C_i, m_i, \pi_{i,\sigma}, \pi_{i,C}, \sigma_{i,O}, ovk_i)$ to the i-th submission query, but for which $ovk = ovk_j$ for some j. Assuming that H'' is collision resistant, which holds in the ROM, but with probability at most $(Q'')^2/q$, if Q'' is the total number of H'' queries made in the game, this failure event implies that (σ_O, h'') is a valid one-time signature-message pair under ovk_j, where $h'' := H''(\pi_\sigma, \pi_C, C, vid, m)$, and distinct from $(\sigma_{j,O}, h''_j)$, where $h''_j := H''(\pi_{j,\sigma}, \pi_{j,C}, C_j, vid_j, m_j)$. Indeed, if $h'' \neq h''_j$ this is clear, and otherwise we must have $\sigma_O \neq \sigma_{j,O}$ since $Sub \neq Sub_j$ and $(\pi_\sigma, \pi_C, C, vid, m) = (\pi_{j,\sigma}, \pi_{j,C}, C_j, vid_j, m_j)$. However, this can only happen with negligible probability according to Assumption 6. More precisely, given the random generators $g, h \in \mathbb{G}_1$ that are common in all the ovk, any forgery against the SUF-CMA scheme allows (tightly) computing the discrete logarithm of h in base g (even if we do not know in advance which ovk_j will be reused), which contradicts the SXDH assumption.

- In Hybrid 6, we bring yet another modification in the way the random oracle answers to the H' queries. For any fresh H' query (ovk, vid) not made by the challenger when answering a submission query, the random oracle defines the output $\mathrm{CRS}'[1]$ so that CRS' induces a Groth–Sahai extractable proof system for multi-scalar multiplication so that witness $x \in \mathbb{Z}_q$ can be extracted as $\hat{g}^x \in \mathbb{G}_2$. We stress that if the challenger does not abort because the failure event introduced in Hybrid 4 does not occur, this change never affects the distribution of $\mathrm{CRS}'_i[1] = H'(ovk_i, vid)$ for any index i so that CRS'_i remains in the simulatable mode. According to Assumption 4, the adversary may only change his behavior with respect to Hybrid 5 with negligible probability. More precisely, a single reduction from the SXDH problem will make this change indistinguishable thanks to the random-self reducibility of SXDH.
- In Hybrid 7, the challenger picks a random pair (sid, sid') of exponents for each honest users. It still computes all the $pk = g^{sid} \in PK_{honest}$, but in all the submission queries, it no longer computes the token C as $H(vid)^{sid}$ but as $H(vid)^{sid'}$. According to Assumption 5, this change is indistinguishable in the ROM. More precisely, the distinguishing advantage between these two hybrids is at most the advantage of distinguishing an SXDH instance. Indeed, let (g, pk_0, h_0, C_0) be this instance, where $pk = g^{sid_0}$ and $C_0 = h_0^{sid_0 + bz}$, for random scalars $sid_0, z \in \mathbb{Z}_q$ and a random bit b to guess. Any honest pk_i is generated as $g^{\alpha_i} pk_0^{\beta_i}$ for random $\alpha_i, \beta_i \leftarrow \mathbb{Z}_q$ (and the proof is simulated in the registration as introduced since Hybrid 1). All the H queries on input vid_j are now answers with $h_0^{\gamma_j}$, for a random $\gamma_j \in \mathbb{Z}_q$ per vid_j. When answering a submission for the i-th honest user, the challenger computes the token $C_i(vid_j) = (h_0^{\alpha_i} C_0^{\beta_i})^{\gamma_j}$ (and simulates the proof as introduced since Hybrid 3). We note that H outputs are still random, which provides no change. Now, we can see that $pk_i = g^{\alpha_i} pk_0^{\beta_i} = g^{sid_i}$ for unknown $sid_i = \alpha_i + sid_0 \beta_i$, and $C_i(vid_j) = (h_0^{\alpha_i} C_0^{\beta_i})^{\gamma_j} = H(vid_j)^{sid'_i}$ for unknown $sid'_i = sid_i + bz\beta_i$. Depending on the bit b, the distribution is either the one of Hybrid 6 or Hybrid 7.
- In Hybrid 8, we introduce a third failure event which causes the challenger to abort the game and outputs `reject` if it occurs. At the end of the game, from any valid submission computed by the adversary, the challenger takes c_{pk} in π_σ and the valid proof π_C, and extracts $\hat{g}^{sid}, \hat{g}^{\rho_1}, \hat{g}^{\rho_2}$ (where $\rho_i = r_i^{pk}$ in the honest case, for $i = 1, 2$) for some unknown exponents such that $e(C, \hat{g}) = (H(vid), \hat{g}^{sid})$ as well as $e(c_{pk}[1], \hat{g}) = e(g, \hat{g}^{sid}) e(g_{11}, \hat{g}^{\rho_1}) e(g_{21}, \hat{g}^{\rho_2})$ and $e(c_{pk}[2], \hat{g}) = e(g_{12}, \hat{g}^{\rho_1}) e(g_{22}, \hat{g}^{\rho_2})$. The last two equations imply $e(pk, \hat{g}) = e(g, \hat{g}^{sid})$ from the extracting key of CRS_1, which, with the first one gives $C = H(vid)^{sid}$ for the same scalar. We recall that if the challenger does not already abort due to the failure events introduced earlier, the associated ovk has never been used when emulating an honest user during a submission and that the CRS′ on such adversarially chosen input is indeed extractable from Hybrid 6. By definition, the new failure event occurs if $pk \in PK_{honest}$. Under Assumption 5, this happens only with negligible probability. More precisely, given an SXDH instance as in the previous reduction, and simulating all the honest users' computation in the same manner, if some pk_i equals an extracted pk (from a necessarily malicious submission) we can tell whether (g, pk_0, h_0, C_0) is random or not. Indeed, thanks to $\hat{g}^{sid} = \hat{g}^{sid_i}$, we check if $e(H(vid_j), \hat{g}^{sid_i}) = e(C_i(vid_j), \hat{g})$ for any vid_j. If it holds, $b = 0$, else, $b = 1$, with probability one (contradicting the SXDH assumption).
- In Hybrid 9, the challenger emulates Hybrid 8, but it also extracts all the group elements from any valid proof π_σ from any valid submission output by the adversary. This extraction works with probability one from Hybrid 2, allowing to compute valid signatures σ_{RA} on (pk, ID) and σ_{SA} on (ID, vid). If the challenger does not abort by applying the rules of the previous failure events, we necessarily have $pk \notin PK_{honest}$. Therefore, if the winning conditions $|S| > |L \cap L_{corrupt}|$ and $(C, m, \pi), (C', m', \pi') \in S \Rightarrow C \neq C'$ hold that means that there is a token $C = H(vid)^{sid}$ for some unknown sid with $pk = g^{sid} \notin PK_{corrupt}$, that is, it has not been registered. This is due to the perfect soundness of the proofs. In that case, the challenger aborts and outputs `reject`. It is easy to see that the probability to reject in Hybrid 9 while Hybrid 8 would accept is bounded by the probability that σ_{RA} or σ_{SA} is a forgery which is negligible by Assumption 1.

It is easy to see that the adversary is never deemed successful in Hybrid 9 since the challenger outputs `reject` in all cases. In summary, the adversary's advantage is bounded by $7\epsilon_{SXDH} + (Q')^2/q^2 + (Q'')^2/q + 2\epsilon_{SPS}$.