# Anonymous Tokens with Public Metadata and Applications to Private Contact Tracing

Tjerand Silde[1] and Martin Strand[2]

[1] Department of Mathematical Sciences
Norwegian University of Science and Technology – NTNU
tjerand.silde@ntnu.no
[2] Norwegian Defence Research Establishment – FFI
martin.strand@ffi.no

**Abstract.** Anonymous single-use tokens have seen recent applications in private Internet browsing and anonymous statistics collection. We develop new schemes in order to include public metadata such as expiration dates for tokens. This inclusion enables planned mass revocation of tokens without distributing new keys, which for natural instantiations can give 77 % and 90 % amortized traffic savings compared to Privacy Pass (Davidson *et al.*, 2018) and DIT: De-Identified Authenticated Telemetry at Scale (Huang *et al.*, 2021), respectively. By transforming the public key, we are able to append public metadata to several existing protocols essentially without increasing computation or communication.
Additional contributions include expanded definitions, a more complete framework for anonymous single-use tokens and a description of how anonymous tokens can improve the privacy in dp$^3$t-like digital contact tracing applications. We also extend the protocol to create efficient and conceptually simple tokens with both public and private metadata, and tokens with public metadata and public verifiability from pairings.

**Keywords:** anonymous tokens · public metadata · contact tracing

## 1 Introduction

Anonymous credentials have been an active research area since the 1980's [Cha82, Cha83], involving schemes such as blind signatures, partially blind signatures, anonymous tokens, attribute-based credentials, group signatures, ring signatures etc. This enables more complex systems for e.g., electronic cash or electronic voting, but also, to protect the privacy of the users in chat applications like Signal.

Recent work by Davidson *et al.* [DGS+18] presents a very practical protocol, named Privacy Pass [DGS+], for anonymous single-use tokens. This protocol allows users to browse anonymously, e.g., using Tor, without having to solve a CAPTCHA every time they visit a website. Privacy Pass gives the user a set of randomized tokens whenever they solve a CAPTCHA, which they then later can redeem instead of solving a new CAPTCHA. This improves the usability of anonymous browsing. It also gives protection against spam, prevents DDoS

attacks and provides fraud resistance without the need for cross-site tracking or fingerprinting. However, the only way to expire or revoke batches of unspent tokens is by replacing the private-public key pair in a trusted way, which is impractical [Dav21].

Privacy Pass has gained a lot of attention, and is currently being integrated to improve privacy in several applications, e.g., for private file storage[3] and for basic attention tokens (BATs) in the Brave browser[4]. It can also be used for private click measurement when making a purchase or signing up for a service[5].

Facebook uses partially blind signatures for combating fraud [IT21], and they have developed an extension of Privacy Pass called DIT: De-Identified Authenticated Telemetry at Scale [HIJ+21], which is used for privately collecting client-side telemetry from WhatsApp. DIT requires daily key-rotation to prevent DoS attacks, which led to the development of an attribute-based verifiable oblivious pseudorandom function for transparent key-rotation.

The IETF is currently standardizing Privacy Pass [Int21], while Trust Token [Wor21] is currently being standardized by the World Wide Web Consortium. Both standardization processes mention private and public metadata, in addition to public verifiability, as desirable extensions to the Privacy Pass protocol. Public metadata allows for more efficient key-rotation, and opens for applications using public labeling and public anonymity sets, while private metadata allows for allow/deny lists, rate-limiting, or trust-indication. Public verifiability allows for outsourcing signing or verification of tokens.

Kreuter *et al.* [KLOR20a] gave the first construction of anonymous tokens with private metadata, while we give the first construction with public metadata. Our construction can also be combined with private metadata or public verifiability.

Privacy Pass guarantees anonymity for all tokens generated by the same key. The addition of any metadata reduces the anonymity set. We have designed the protocol in such a way that the user and the signer must agree on the metadata. Any application should restrict its use of metadata to generic, predefined values that would otherwise have triggered a change of keys, e.g., expiry dates. Client software should validate that the metadata is in accordance to the policy, and reject any malformed tokens. Furthermore, private metadata bits also reduces the anonymity set. Our protocol can easily be extended to include more than one private metadata bit, but this must be done with great care, as it opens for secretly tracking smaller sets of individual users.

Independently of this work, Tyagi *et al.* [TCR+21] have proposed a similar construction to include public metadata, along with a novel hardness assumption and a reduction to a more conventional problem, to be used in partially oblivious pseudo-random functions. We discuss their work further in Sections 1.5 and 4.

[3] PrivateStorage: medium.com/least-authority/the-path-from-s4-to-privatestorage-ae9d4a10b2ae.

[4] Brave: github.com/brave/brave-browser/wiki/Security-and-privacy-model-for-ad-confirmations.

[5] Private Click Measurement: privacycg.github.io/private-click-measurement.

### 1.1 Background

Adopting the terminology from Privacy Pass [DGS+18], we have the following informal architecture. A *user* asks the *signer* for a one-time anonymous token in a *signing phase*, later to be redeemed to a *verifier* in a *redemption phase*.

The literature provides many flavors of anonymous credentials. They all come with some minimal requirements with respect to security, and have a variation of desirable properties for practical applications.

**Unlinkability and Unforgeability.** To ensure privacy of users, the anonymous token protocol must make sure that the information being transferred in the attestation phase cannot be correlated with the token being received in the redemption phase. This is called unlinkability. To ensure the system's integrity, the anonymous token protocol must make sure that users cannot forge tokens, even after receiving valid tokens from the attestation server. This is called unforgeability. These are the minimal requirements for anonymity and integrity, and have been handled starting from the first classical constructions of blind signatures starting with David Chaum in the early 1980's [Cha82, Cha83] and subsequent work on anonymous credentials [Oka93, PS96, CL01, CL03, CL04, CHL05, CG08, GMS10, BL13, CMZ14, CPZ20].

**Underlying Primitives.** Anonymous token protocols can be built from a variety of cryptographic primitives and assumptions, e.g., factoring [AF96, CL01, CL03], discrete logarithms [Oka93, AO00, WSMZ06, DGS+18, KLOR20a] or bilinear pairings [ZSS03, CHYC05, CZMS06, BPV12]. Protocols based on elliptic curves are the most efficient, both in terms of size and timings, while other primitives might more easily provide correctness and verifiability.

**Verifiability and Key-Sharing.** In situations where the same party is both attesting and redeeming tokens [DGS+18, KLOR20a], it is natural for the server(s) to share a key. In the designated verifier setting, it is necessary for the attestation server to provide zero-knowledge proofs to ensure that a token is honestly generated [DGS+18, KLOR20a], while pairings can provide public verifiability directly [ZSS03].

**Key-Rotation and Token-Revocation.** To avoid misuse, it is important to have a mechanism to efficiently expire or revoke batches of tokens. This may be useful for rate limiting to avoid denial-of-service attacks, or to protect users from credential stuffing [DGS+18, TPY+19, HIJ+21]. In Privacy Pass [DGS+18], this is solved by infrequent key-rotation where a few public keys are available at a public endpoint. DIT [HIJ+21], which rotates their keys every day, solves it by using a new attribute-based verifiable oblivious pseudorandom function (VOPRF). However, both solutions are inconvenient in practice, and add significant overhead for validating the public keys.

**Rounds of Interaction.** Blind signatures and anonymous tokens can be attested in only one round of communication, which is optimal. This saves time and computation for both the client and the server, and the parties does not need to keep a state. However, the only partially blind signatures achieving one round of interaction are based on bilinear pairings or factoring [AF96], while protocols based on discrete logarithms [AO00, WSMZ06] needs two rounds. We note that there is no one-round single-use anonymous token protocol with efficient revocation in the literature with security based on elliptic curve discrete logarithms without pairings before this work. Popular schemes such as [CL04, BL13] require at least two rounds of communication.

### 1.2 Our Contribution

Our contribution in this paper is threefold: first, we present new definitions and a new framework for anonymous tokens – extending the work by Kreuter *et al.* [KLOR20a] – to also consider public metadata and/or public verifiability. Secondly, we present three efficient protocols for anonymous tokens with efficient batched revocation: 1) Privacy Pass [DGS+18] with public metadata, 2) Kreuter *et al.* [KLOR20a] with public and private metadata, and 3) a Privacy Pass inspired protocol using pairings to satisfy public verifiability while including public metadata. Thirdly, we present contact tracing as a new and important application for anonymous single-use tokens, and discuss the implementation of Privacy Pass used in the Norwegian contact tracing app Smittestopp to improve users' privacy.

**Updated Definitions and New Framework.** Several works have asked for efficient batched revocation of anonymous tokens without key-rotation [DGS+18, Dav21]. Additionally, there is a need for anonymous tokens with public verifiability [Wor21], so that token generation can be delegated, and verification can be performed locally for token redemption. We provide updated definitions for all of these cases: designated verifier anonymous tokens with or without public and/or private metadata and public verifier anonymous tokens with and without public metadata. Details can be found in Section 3.

**Anonymous Tokens with Public Metadata.** We present the first anonymous tokens protocols with efficient batched revocation, meaning that the protocol only requires one round of communication based on lightweight primitives and that we avoid key-rotation. The key insight in our protocol is conceptually very simple: all parties locally update the public key based on the hash of the public metadata, and then execute the protocols with respect to the new key pair. The main challenge is to sign tokens in a way that does not allow the user to forge tokens initially signed under metadata $\mathsf{md}$ to be valid under metadata $\mathsf{md}'$ instead. Let $k$ be the secret key and let $d = \mathtt{H}(\mathsf{md})$ be the hash of the metadata. Our solution, inspired by Zhang *et al.* [ZSS03], is to use the inverse $e = (d+k)^{-1}$

as the new signing key. This allows us to replace the secret keys in the previous protocols in a modular way.

Furthermore, to avoid subliminal channels, the signer needs to prove that the signed token is computed correctly. This is easily solved for Privacy Pass [DGS+18]. In the original protocol they use a zero-knowledge protocol to prove, given generator $G$, public key $K = [k]G$, blinded token $T'$ and signed token $W' = [k]T'$, the equality of discrete logarithms $\log_G K = k = \log_{T'} W'$ to ensure correctness. In our updated protocol, including metadata `md`, updated public key $U = [d]G + K$ and signed token $W' = [e]T'$, we prove the equality of discrete logarithms $\log_G U = d + k = \log_{W'} T'$ to ensure correctness.

However, it is not as easy to ensure correctness in the extended version of the protocol by Kreuter *et al.* [KLOR20a] including both public and private metadata. We solve this by combining an `OR`-proof with two `AND`-proofs to make sure that the correct key is used. Further improvement is an open problem.

Next, we give a protocol based on pairings. The protocol is an adapted version of the partially blind signatures by Zhang *et al.* [ZSS03], where we tweak it into the same structure as Privacy Pass. We note that the communication in the protocol is the same, but in addition to get a more streamlined protocol structure, we also allow for more efficient instantiation in practice using the BLS12-381 pairing [BLS03]. Ideally, we would like to avoid pairings altogether, but this seems necessary in practice. See more details about the protocols in Section 4.

Finally, we detail the communication efficiency of the protocols in Section 5, and compare our constructions with the current state of the art with respect to efficient batched revocation in Table 1. We show that our protocols are much more efficient in practice. We also make a concrete comparison with DIT [HIJ+21] for collecting telemetry-data from WhatsApp, and show that our protocol in Figure 6 would decrease the size of the signed token in a natural setting by 90 %, saving the Facebook servers up to 1.7 TB of communication every day.

**More Private Contact Tracing.** Many countries have recently developed contact tracing apps as one of the measurements to battle the ongoing pandemic. These apps are inherently storing sensitive information about the user, e.g., the users' location graph and social graph. To avoid large, centralized databases with such sensitive information about a large portion of a country's adult population, most apps are based on the decentralized Google/Apple Exposure Notification System (ENS). However, there are still privacy issues with regards to uploading the randomized exposure keys to the central server, as the user would have to identify themselves to ensure that only people who have tested positive for COVID-19 are able to upload keys. We implemented Privacy Pass into the Norwegian contact tracing app to improve the users' privacy. Our code is published at github.com/HenrikWM/anonymous-tokens, and the Norwegian Institute of Public Health (NIPH) have made the source code for the contact tracing infras-

tructure publicly available[6]. We present more details about the contact tracing infrastructure and improvements in Section 6.

### 1.3 Comparison to Anonymous Credentials

There is a long line of research on more generalized anonymous credentials with features such as multi-show, multi-attributes, and revocability – in addition to the mandatory unlinkability and unforgeability – that allow one to encode expiration dates as attributes.

However, generalized anonymous credentials often depends on stronger assumptions, e.g., strong RSA [CL01, CV02, CL03, CHL05, CG08], strong Diffie-Hellman [AMO08] or DL assumptions in bilinear groups [CL04, HS21]. Some schemes only depend on DDH [BL13, PZ13, CMZ14, CPZ20], but these schemes require larger messages in general. In conclusion, generalized anonymous credentials inherently impose larger parameters, more rounds of communication and less efficient protocols in practice, resulting in thousands of bits on communication over multiple rounds.

Finally, more general and complex anonymous credentials make these schemes less suited for use in simpler single-use systems with many users, which is the case in our setting. We want to minimize the rounds of communication and data being sent, in addition to minimizing the local computation and the local state. Hence, we only compare to one-round single-use efficiently revocable anonymous credentials with minimal communication in Section 5.

### 1.4 Related work

Our work achieving designated verification and public metadata extends a long line of publications. Freedman *et al.* [FIPR05] introduced oblivious pseudorandom functions, and Jarecki *et al.* [JKK14, JKX18] gave an efficient instantiation based on DDH in the random oracle model. Papadopoulos *et al.* [PWH+17] gave a verifiable PRF from elliptic curves, and Burns *et al.* [BMR+17] gave an oblivious PRF from elliptic curves. Privacy Pass combined these results with an extended version of the Chaum-Pedersen zero-knowledge protocol [CP93] given by Henry and Goldberg [HG13, Hen14] to prove knowledge of batches of elements having the same discrete logarithm, and Kreuter *et al.* [KLOR20a] added private metadata to Privacy Pass. In a concurrent work, Tyagi *et al.* [TCR+21] recently extended this line of works to partially oblivious PRFs.

To achieve public verifiability we use parings, inspired by the seminal work of Boneh *et al.* [BLS01] for short and efficient signatures and a series of constructions of (partially) blind signatures based on pairings [ZSS03, Bol03, CHYC05, CZMS06, CKS09, BPV12, FHS15, FHKS16].

[6] NIPH: github.com/folkehelseinstituttet/?q=Smittestopp.

### 1.5 Chronology

As we report on both an implementation and new protocols, we believe it can be helpful to lay out the chronology of this work to separate the contributions.

Mid-October 2020, the authors were made aware of a potential privacy weakness in Norway's upcoming second COVID-19 contact tracing app Smittestopp. The first iteration had been stopped by the Norwegian Data Protection Agency in June, due to privacy concerns following from lack of data minimization. The new app had a set launch date in December.

The issue was that the verification service would collect IDs in order to automatically verify the infection status, and then send a token to the app which could then be used for uploading exposure keys. This token would create a hard link between an ID-based service and the rest of the system, in which the users are assumed to be anonymous.

Within a few days, we suggested using Privacy Pass in order to remove this link. Due to lack of capacity, our proposal was acknowledged, but we were asked to provide the code. We teamed up with Henrik Walker Moe to implement Privacy Pass in C#, and our implementation was eventually accepted into Smittestopp along with an improvised solution to rotate keys every three days.

Motivated by this process and the last-minute improvisation, we expanded the original Privacy Pass protocol to deal with the issues of key-rotation and revocation. Our initial manuscript was posted on ePrint February 24th, 2021. We were then made aware of a complication to the security proof, which was originally from the work by Zhang *et al.* [ZSS03]. A correct proof was posted on ePrint by Tyagi *et al.* [TCR+21] June 24th, 2021. The primary separation between these two manuscripts are that we were the first to present this protocol along with its variations, while Tyagi *et al.* present a correct proof. We also present the protocols in a way that is compatible to previous work. In this sense, these works complement each other.

The new protocol has not been implemented in Smittestopp. This is due to lack of further development of the app, and we do not expect any major changes to be accepted into the codebase at this stage.

## 2 Preliminaries

We assume that the reader is familiar with the basics of elliptic curve cryptography. To fix notation, let $q$ be a prime and let $\mathbb{F}_{q^\ell}$ for some $\ell > 0$ be a field of characteristic $q$. Let $E$ be all points $(x, y)$ that satisfy the elliptic curve equation $y^2 = x^3 + ax + b$ in the algebraic closure of $\mathbb{F}_{q^\ell}$, and let $E(\mathbb{F}_{q^\ell})$ denote the set of all such points in $\mathbb{F}_{q^\ell} \times \mathbb{F}_{q^\ell}$ along with the point at infinity $\mathcal{O}$. By abuse of notation, we often let $E$ be a group of order $p$ inside $E(\mathbb{F}_{q^\ell})$. Define the group law in the usual additive way. In particular, let $[m] : E \to E$ be the multiplication-by-$m$ map, which takes the same role as exponentiation in multiplicative groups. Now follows a brief discussion of the Chosen-Target Gap Diffie-Hellman problem and some zero-knowledge proofs we will need as primitives.

### 2.1 The One-More Gap Strong Diffie-Hellman Problem

The strong Diffie-Hellman problem was introduced by Boneh and Boyen [BB04]. Given a sequence $g, g^x, g^{x^2}, \ldots, g^{x^q}$ from a group $\mathbb{G}$ of prime order $p$, output a pair $(c, g^{(x+c)^{-1}})$ with $c \in \mathbb{Z}_p$. We now present a variant of this game: the adversary must commit to fixed set of candidates $\{c_i\}$, and may then query an oracle for $B^{(\mathsf{sk}+c_i)^{-1}}$ for arbitrary $B$, along with an oracle for the decision variant. The adversary wins if it can present $\ell + 1$ correct tuples for a chosen $c_i$, but only having queried $\ell$ or less. The details are presented in Figure 1. The definition and game is due to Tyagi *et al.* [TCR+21].

| Game $(m,n)$-OM-GAP-SDHI$_{\mathsf{Gen},\mathcal{A},\ell}(\lambda)$ | Oracle SDH$(B,i)$ |
|---|---|
| $(\mathbb{G}, p, g) \leftarrow \mathsf{Gen}(1^\lambda)$ | **if** $i \notin [n]$ **then** |
| $\mathsf{sk} \leftarrow\!\!\$\ \mathbb{Z}_p$ | **return** $\perp$ |
| $(y_i)_{i\in[m]} \leftarrow\!\!\$\ \mathbb{Z}_p^m$ | $q_i := q_i + 1$ |
| $(\mathsf{st}_\mathcal{A}, (c_i)_{i\in[n]}) \leftarrow \mathcal{A}_1(p, \mathbb{G})$ | $Z := B^{(\mathsf{sk}+c_i)^{-1}}$ |
| $(\gamma, (Z_i, \alpha_i)_{i\in[\ell+1]}) \leftarrow \mathcal{A}_2^{\text{SDH},\text{SDDH}}\left(g, g^{\mathsf{sk}}, (g^{y_i})_{i\in[m]}; \mathsf{st}_\mathcal{A}\right)$ | **return** $Z$ |
| **if** $q_\gamma \le \ell$ **and** $(i \neq j \to \alpha_i \neq \alpha_j)$ **then** | Oracle SDDH$(Y,Z,i)$ |
| **return** $(Z_i)_{i\in[\ell+1]} = \left(g^{y_{\alpha_i}(\mathsf{sk}+c_\gamma)^{-1}}\right)_{i\in[\ell+1]}$ | **return** $Z = Y^{(\mathsf{sk}+c_i)^{-1}}$ |

**Fig. 1.** The one-more gap strong inversion Diffie-Hellman security game.

**Definition 1 ($(m,n)$-One-More Gap Strong Inversion Diffie-Hellman).** *Let $m, n$ be natural numbers, and let $\mathbb{G}$ be a cyclic group of order $p$ with generator $g$ produced by the algorithm $\mathsf{Gen}(1^\lambda)$. Let $(m,n)$-OM-GAP-SDHI be the game defined in Figure 1. $(m,n)$-One-More Gap Strong Diffie-Hellman Inversion holds for $\mathbb{G}$ if for any PPT adversary $\mathcal{A}$ and any $\ell \ge 0$,*

$$\mathsf{Adv}^{\text{om-gap-sdhi}}_{\mathsf{Gen},\mathcal{A},\ell}(\lambda) := \Pr[(m,n)\text{-OM-GAP-SDHI}_{\mathsf{Gen},\mathcal{A},\ell}(\lambda) = 1] = \mathsf{negl}(\lambda).$$

Tyagi *et al.* [TCR+21] have proven that this assumption is implied by the much simpler $q$-DL assumption, which asks the adversary to return $x$, given $g, g^x, g^{x^2}, \ldots, g^{x^q}$.

### 2.2 DDH vs. CDH in Pairings

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order, written additively, and let $\mathbb{G}_T$ be another cyclic group of same prime order, written multiplicatively. A bilinear pairing $\hat{e}$ is a map

$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$$

such that the following properties hold:

**Bilinearity** For all $P_1, P_2 \in \mathbb{G}_1$ and $Q_1, Q_2 \in \mathbb{G}_2$, it holds that $\hat{e}(P_1+P_2, Q_1) = \hat{e}(P_1, Q_1)\hat{e}(P_2, Q_1)$ and $\hat{e}(P_1, Q_1 + Q_2) = \hat{e}(P_1, Q_1)e(P_1, Q_2)$.
**Non-degeneracy** For all $P \neq \mathcal{O}$, $\hat{e}(P, P) \neq 1$.
**Computability** $\hat{e}$ can be efficiently computed.

The bilinearity property implies that for scalars $a, b$, we have $\hat{e}([a]P, [b]Q) = \hat{e}(P, Q)^{ab}$, which is the crucial property used for verification.

Bilinear maps lend themselves to a variant of the well-known Diffie-Hellman problem, the Chosen-Target Gap Diffie-Hellman problem [BNPS02]. Even if the adversary is given oracle access to $\ell$ instances of the Computational Diffie-Hellman (CDH) problem and arbitrary many queries to a Decision Diffie-Hellman (DDH) oracle, it should still not be able to compute the final Diffie-Hellman instance $\ell + 1$. We repeat the game and definition by Kreuter *et al.* [KLOR20a].

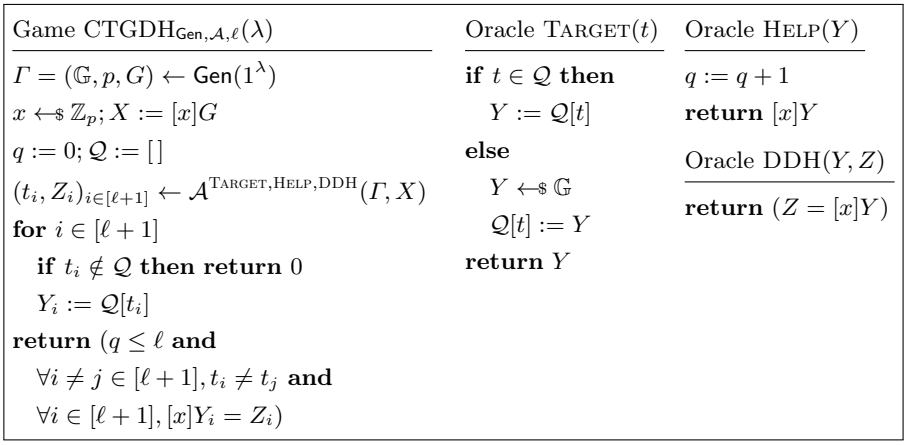


**Fig. 2.** The Chosen-target gap Diffie-Hellman security game.

**Definition 2 (Chosen-Target Gap Diffie-Hellman).** *Let $\mathbb{G}$ be a cyclic group of order $p$ with generator $G$ produced by the algorithm $\mathsf{Gen}(1^\lambda)$. Let* CTGDH *be the game defined in Figure 2. Chosen-Target Gap Diffie-Hellman holds for $\mathbb{G}$ if for any* PPT *adversary $\mathcal{A}$ and any $\ell \geq 0$,*

$$\mathsf{Adv}^{\text{ctgdh}}_{\mathsf{Gen},\mathcal{A},\ell}(\lambda) := \Pr[\text{CTGDH}_{\mathsf{Gen},\mathcal{A},\ell}(\lambda) = 1] = \mathsf{negl}(\lambda).$$

### 2.3 Proof of Equal Discrete Logs

Chaum and Pedersen [CP93] introduced an elegant honest-verifier zero-knowledge protocol to prove that two group elements have the same discrete logarithm relative to their respective bases, $\log_G K = k = \log_T W$. We describe the protocol loosely to ensure the reader is familiar with the idea. Let $\mathbb{G}$ be a cyclic group

of prime order $p$ with independent generators $G$ and $T$, and let $K := [k]G$, $W := [k]T$ where $k$ is a scalar private to the prover $\mathcal{P}$.

$\mathcal{P}$**.1** Choose a random scalar $r$ in the underlying field, compute $A := [r]G$ and $B := [r]T$, and send $(A, B)$ to $\mathcal{V}$.
$\mathcal{V}$**.1** Choose a random challenge $c$ modulo $p$ and send it to $\mathcal{P}$.
$\mathcal{P}$**.2** Compute the response $z := r - ck$ modulo $p$, and then send $z$ to $\mathcal{V}$.
$\mathcal{V}$**.2** Verify that $A = [z]G + [c]K$ and $B = [z]T + [c]W$.

This protocol satisfies unconditional special soundness and special honest-verifier zero-knowledge. One can make the protocol non-interactive by applying the Fiat-Shamir transformation [FS87]. The prover queries the oracle on the tuple $(\mathbb{G}, G, T, K, W, A, B)$. In addition, one can reduce communication by sending the oracle response $c$ instead of $(A, B)$, and modifying the final verification step to querying the oracle on $(\mathbb{G}, G, T, K, W, [z]G + [c]K, [z]T + [c]W)$, and then verify that it indeed returns $c$. We will use a shorthand notation to refer to this proof as $\Pi_{\text{DLEQ}}(G, T, K, W; k)$, meaning that $\log_G([k]G) = \log_W T$.

The proof can be batched for many instances with respect to the same secret scalar using the techniques by Henry [Hen14] as showed in [DGS+18, Section 3.2.1].

### 2.4 AND-Proof of Equal Discrete Logs

Let $\mathbb{G}$ be an additive group of prime order $p$ with independent generators $G, H, T, S$, and let $K := [k_0]G + [k_1]H$ and $V := [k_0]T + [k_1]S$, where $k_0, k_1$ are scalars private to the prover $\mathcal{P}$. We want to prove that $V$ is correctly computed with respect to $T$ and $S$ using the same secret scalars as $K$ with respect to $G$ and $H$. We present a simple protocol to prove this relation, by essentially computing two Chaum-Pedersen proofs in parallel.

$\mathcal{P}$**.1** Choose two random scalars $r_0, r_1$ modulo $p$. Compute $A := [r_0]G + [r_1]H$, $B := [r_0]T + [r_1]S$, and send $(A, B)$ to $\mathcal{V}$.
$\mathcal{V}$**.1** Choose a random challenge $c$ modulo $p$ and send it to $\mathcal{P}$.
$\mathcal{P}$**.2** Compute $z_0 := r_0 - ck_0$ and $z_1 := r_1 - ck_1$ modulo $p$ and send them to $\mathcal{V}$.
$\mathcal{V}$**.2** Verify that $A = [c]K + [z_0]G + [z_1]H$ and $B = [c]V + [z_0]T + [z_1]S$.

It is straightforward to verify that this is a sigma protocol with special soundness and special honest-verifier zero-knowledge. As above, we can apply the Fiat-Shamir [FS87] transformation to get a non-interactive protocol. We will refer to this proof as $\Pi_{\text{DLEQ2}}(G, H, T, S, K, V; k_0, k_1)$.

### 2.5 OR-Proof of Equal Discrete Logs

We present the honest-verifier zero-knowledge `OR`-proof of equal discrete logarithms instantiated by Kreuter et al. [KLOR20b, Appendix B]. Let $\mathbb{G}$ be a cyclic group of prime order $p$ with generators $G, H, T, S$, and let $V_0 := [e_{0,0}]G + [e_{0,1}]H$ and $V_1 := [e_{1,0}]G + [e_{1,1}]H$, where $e_{i,j}$ are distinct scalars private to the prover $\mathcal{P}$. Furthermore, let $W := [e_{b,0}]T + [e_{b,1}]S$ for $b \in \{0, 1\}$. We want to prove that $W$ is computed using the same secret scalars as either $V_0$ or $V_1$.

$\mathcal{P}$**.1** Choose random scalars $r_0, r_1, c_{b-1}, u_{b-1}, v_{b-1}$ in the underlying field and compute the following:

$$\begin{aligned} A_{b,0} &:= [r_0]G + [r_1]H, \\ A_{b,1} &:= [r_0]T + [r_1]S, \\ A_{1-b,0} &:= [u_{b-1}]G + [v_{b-1}]H - [c_{b-1}]V_{b-1}, \\ A_{1-b,1} &:= [u_{b-1}]T + [v_{b-1}]S - [c_{b-1}]W. \end{aligned}$$

Finally, send $(A_{0,0}, A_{0,1}A_{1,0}, A_{1,1})$ to $\mathcal{V}$.

$\mathcal{V}$**.1** Choose a random challenge $c$ modulo $p$ and send it to $\mathcal{P}$.

$\mathcal{P}$**.2** Compute the responses

$$c_b := c - c_{1-b}, \quad u_b := r_0 + c_b e_{b,0}, \quad v_b := r_1 + c_b e_{b,1},$$

modulo $p$. Send $(c_i, u_i, v_i)_{i=0,1}$ to $\mathcal{V}$.

$\mathcal{V}$**.2** Verify that $c = c_0 + c_1$ and that

$$\begin{aligned} A_{0,0} &= [u_0]G + [v_0]H - [c_0]V_0, \\ A_{0,1} &= [u_0]T + [v_0]S - [c_0]W, \\ A_{1,0} &= [u_1]G + [v_1]H - [c_1]V_1, \\ A_{1,1} &= [u_1]T + [v_1]S - [c_1]W. \end{aligned}$$

We can make the proof non-interactive using Fiat-Shamir [FS87] like above. We will refer to this protocol as $\Pi_{\text{DLEQOR2}}(G, H, T, S, V_0, V_1, W; e_{b,0}, e_{b,1})$.

$\Pi_{\text{DLEQOR2}}$ can be batched for many instances with respect to the same secret scalars using the techniques by Henry [Hen14] as shown in [KLOR20a, Appendix B.1].

## 3 Definitions for Anonymous Tokens

Anonymous tokens as used in Privacy Pass are conceptually simple: both issuance and verification require the private key, and the final token is uniquely determined by the token seed $t$ and the private key. Kreuter *et al.* [KLOR20a] extended this notion by adding a private bit in the token. We further extend the definition in two different directions: we want to add public metadata, and we want to make the token publicly verifiable. Now, private bits do not make immediate sense in the context of a publicly verifiable token scheme, but public metadata can be relevant in both settings.

The metadata can for instance be used to indicate an expiry date, replacing the need for frequent key rotation in certain applications [HIJ+21] as we discussed in Section 1.1. We model it as a value that the user and issuer must agree upon, which should restrict the issuer from using arbitrary, identifiable values.

Lending terminology from programming, we would like the definition to provide backwards compatibility, and handle the notational incompatibility between private and public verifiability. To this end, we imitate the notion of [optional

arguments] from programming. The notation $\mathsf{vk}|\mathsf{sk}$ is meant as "at least one of the public or the secret key". We align our definitions as close as possible to those by Kreuter *et al.* [KLOR20a].

**Definition 3 (Anonymous tokens).** *An anonymous token scheme with zero or more of* ***private metadata bit****,* ***public metadata****, or* ***public verifiability*** *consists of the following algorithms:*

- $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{AT.Setup}(1^\lambda)$*, the setup algorithm that takes as input the security parameter* $\lambda$ *in unary form, and returns a common reference string* $\mathsf{crs}$ *and trapdoor* $\mathsf{td}$*. All the remaining algorithms take* $\mathsf{crs}$ *as their first input.*

- $(\mathsf{pp}, \mathsf{sk}, [\mathsf{vk}]) \leftarrow \mathsf{AT.KGen}(\mathsf{crs})$*, the key generation algorithm that generates a signing key* $\mathsf{sk}$ *and optionally a verification key* $\mathsf{vk}$ *along with public parameters* $\mathsf{pp}$*. All the remaining algorithms take* $\mathsf{pp}$ *as their second input.*

- $\sigma \leftarrow \langle \mathsf{AT.User}(\mathsf{pp}, [\mathsf{vk}], t, [\mathsf{md}]), \mathsf{AT.Sign}(\mathsf{sk}, [\mathsf{md}], [b]) \rangle$*, the token issuance protocol, which involves interactive algorithms* $\mathsf{AT.User}$ *and* $\mathsf{AT.Sign}$*. The user algorithm takes as input values the public parameters and the token seed* $t \in \{0,1\}^\lambda$*, and potentially the verification key* $\mathsf{vk}$ *and the public metadata* $\mathsf{md}$*. The signing algorithm takes the private key* $\mathsf{sk}$ *and potentially metadata* $\mathsf{md}$ *and the private bit* $b$*. At the end of the interaction, the issuer outputs nothing, while the user outputs* $\sigma$*, or* $\perp$ *to indicate error.*

- $bool \leftarrow \mathsf{AT.Vf}(\mathsf{vk}|\mathsf{sk}, t, [\mathsf{md}], \sigma)$*, the verification algorithm that takes as input either the public verification key* $\mathsf{vk}$ *or the private key* $\mathsf{sk}$*, a token seed* $t$*, metadata* $\mathsf{md}$ *and the signature* $\sigma$*. It returns true if the token was valid.*

- $[\mathsf{ind} \leftarrow \mathsf{AT.ReadBit}(\mathsf{sk}, t, [\mathsf{md}], \sigma)]$*, the private bit extraction algorithm that takes as input the private key* $\mathsf{sk}$ *and token* $(t, [\mathsf{md}], \sigma)$*. It returns an indicator* $\mathsf{ind} \in \{\perp, 0, 1\}$ *which is either the private bit, or* $\perp$*.*

The notation of the above definition should be interpreted in a global sense. If one – for example – wants to use public metadata, it should be included everywhere it is mentioned. This listing then defines the following six notions:

1. With designated verification:
   (a) Anonymous single-use tokens
   (b) Anonymous single-use tokens with private metadata bit
   (c) Anonymous single-use tokens with public metadata
   (d) Anonymous single-use tokens with public and private metadata
2. With public verification:
   (a) Anonymous single-use tokens
   (b) Anonymous single-use tokens with public metadata

Examples of 1a and 1b are well known from previous work [DGS+18,KLOR20a]. A previous example of 2b is known as a partially blind signature scheme [AO00].

We will provide new examples of the last four (2a is implicit in 2b) in Section 4. We collectively refer to all of these as anonymous tokens.

We follow the convention of dividing the interactive protocol $\langle \mathsf{AT.User}, \mathsf{AT.Sign} \rangle$ into the non-interactive algorithms $\mathsf{AT.User}_0$, $\mathsf{AT.Sign}_0$ and $\mathsf{AT.User}_1$.

An anonymous token scheme must satisfy the following properties:

**Definition 4 (Token correctness).** *An anonymous token scheme* $\mathsf{AT}$ *is correct if any honestly generated token verifies. For any honestly generated* $\mathsf{crs}$, $(\mathsf{pp}, \mathsf{sk}, [\mathsf{vk}])$, $t$ *and* $[\mathsf{md}]$,

$$\Pr[\mathsf{AT.Vf}(\mathsf{vk}, t, [\mathsf{md}], \langle \mathsf{AT.User}(\mathsf{pp}, [\mathsf{vk}], t, \mathsf{md}), \mathsf{AT.Sign}(\mathsf{sk}, [\mathsf{md}], [b]) \rangle) = 1] = 1 - \mathsf{negl}(\lambda).$$

We split correctness of the private metadata bit into a separate definition in order to reduce notational clutter. This definition only applies in the private-key setting, and the parameters have been fixed accordingly.

**Definition 5 (Correct private bit).** *An anonymous token scheme* $\mathsf{AT}$ *is correct with respect to private metadata if the correct bit is retrieved successfully:*

$$\Pr[\mathsf{AT.ReadBit}(\mathsf{sk}, t, \langle \mathsf{AT.User}(\mathsf{pp}, t, [\mathsf{md}]), \mathsf{AT.Sign}(\mathsf{sk}, [\mathsf{md}], b) \rangle) = b] = 1 - \mathsf{negl}(\lambda).$$

No adversary should be able to redeem other tokens than those that have been correctly issued. The *one-more unforgeability* notion has become the common notion for anonymous credentials. It allows the adversary to claim $\ell$ tokens from the issuer, and the adversary should not be able to redeem $\ell + 1$ tokens. We require the tokens to be unique with respect to the value of the seed $t$.

| Game $\mathrm{OMUF}_{\mathsf{AT},\mathcal{A},\ell}(\lambda)$ | Oracle $\textsc{Sign}(\mathsf{msg}, [\mathsf{md}], [b])$ |
|---|---|
| $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{AT.Setup}(1^\lambda)$ | $q_{b,\mathsf{md}} := q_{b,\mathsf{md}} + 1$ |
| $(\mathsf{pp}, \mathsf{sk}, [\mathsf{vk}]) \leftarrow \mathsf{AT.KGen}(\mathsf{crs})$ | **return** $\mathsf{AT.Sign}_0(\mathsf{sk}, \mathsf{msg}, [\mathsf{md}], [b])$ |
| **for** $(b \in \{0,1\}, \mathsf{md}), q_{b,\mathsf{md}} := 0$ | Oracle $\textsc{Verify}(t, [\mathsf{md}], \sigma)$ |
| $(t_i, \mathsf{md}_i, \sigma_i)_{i \in [\ell+1]} \leftarrow \mathcal{A}^{\textsc{Sign},\textsc{Verify},\textsc{Read}}(\mathsf{crs}, \mathsf{pp})$ | **return** $\mathsf{AT.Vf}(\mathsf{sk}\|\mathsf{vk}, t, [\mathsf{md}], \sigma)$ |
| **return** $(\forall b \in \{0,1\}\ \forall \mathsf{md}, q_{b,\mathsf{md}} \leq \ell$ **and** | Oracle $\textsc{Read}(t, \sigma)$ |
| $\forall i \neq j$ **in** $[\ell+1]$ $(t_i, \mathsf{md}_i, \sigma_i) \neq (t_j, \mathsf{md}_j, \sigma_j)$ | **return** $\mathsf{AT.ReadBit}(\mathsf{sk}, t, [\mathsf{md}], \sigma)$ |
| **and** $\exists (b, \mathsf{md}) \in \{0,1\} \times \{\mathsf{md}\} : \forall i \in [\ell+1],$ | |
| $\mathsf{AT.ReadBit}(\mathsf{sk}, t_i, \sigma_i) = b$ **and** | |
| $\mathsf{AT.Vf}(\mathsf{sk}\|\mathsf{vk}, t_i, [\mathsf{md}], \sigma_i) = \mathbf{true})$ | |

**Fig. 3.** One-more unforgeability with metadata.

| Game $\text{UNLINK}_{\mathsf{AT},\mathcal{A},m,[b],[\mathsf{md}]}(\lambda)$ | Oracle $\text{USER}_0()$ |
|---|---|
| $(\mathsf{crs},\mathsf{td}) \leftarrow \mathsf{AT.Setup}(1^\lambda)$ | $q_0 := q_0 + 1$ |
| $(\mathsf{st},\mathsf{pp},[\mathsf{vk}]) \leftarrow \mathcal{A}(\mathsf{crs},[b],[\mathsf{md}])$ | $t_{q_0} \leftarrow\!\!\$\ \{0,1\}^\lambda$ |
| $q_0 := 0; q_1 := 0, \mathcal{Q} := \emptyset$ | $(\mathsf{msg}_{q_0},\mathsf{st}_{q_0}) \leftarrow \mathsf{AT.User}_0(\mathsf{pp},[\mathsf{vk}],t_{q_0},[\mathsf{md}'])$ |
| $(\mathsf{st},(\mathsf{msg}_i)_{i\in\mathcal{Q}}) \leftarrow \mathcal{A}^{\text{USER}_0,\text{USER}_1}(\mathsf{st})$ | $\mathcal{Q} := \mathcal{Q} \cup \{q_0\}$ |
| **if** $\mathcal{Q} = \emptyset$ **then return** 0 | **return** $(q_0,\mathsf{msg}_{q_0})$ |
| $j \leftarrow\!\!\$\ \mathcal{Q}; \mathcal{Q} = \mathcal{Q} \setminus \{j\}$ | Oracle $\text{USER}_1(j,\mathsf{msg})$ |
| $\sigma_j \leftarrow \mathsf{AT.User}_1(\mathsf{st}_j,[\mathsf{vk}],\mathsf{msg}_j,[\mathsf{md}])$ | **if** $j \notin \mathcal{Q}$ **then** |
| **for** $i \in \mathcal{Q}$ | **return** $\perp$ |
| $\sigma_i \leftarrow \mathsf{AT.User}_1(\mathsf{st}_i,[\mathsf{vk}],\mathsf{msg}_i,[\mathsf{md}])$ | $\sigma \leftarrow \mathsf{AT.User}_1(\mathsf{st}_j,[\mathsf{vk}],\mathsf{msg},[\mathsf{md}'])$ |
| $\phi \leftarrow\!\!\$\ \mathcal{S}_\mathcal{Q}$ | **if** $\sigma \neq \perp$ **then** |
| $j' \leftarrow \mathcal{A}(\mathsf{st},(t_j,\sigma_j),(t_{\phi(i)},\sigma_{\phi(i)})_{i\in\mathcal{Q}})$ | $\mathcal{Q} := \mathcal{Q} \setminus \{j\}$ |
| **return** $q_0 - q_1 \geq m$ **and** $j' = j$ | $q_1 := q_1 + 1$ |
| | **return** $\sigma$ |

**Fig. 4.** Public-key unlinkability with fixed metadata. If $X$ is a set, then $\mathcal{S}_X$ is the symmetric group of $X$.

**Definition 6 (One-more unforgeability).** *An anonymous token scheme* $\mathsf{AT}$ *is one-more unforgeable if for any* PPT *adversary* $\mathcal{A}$*, and any* $\ell \geq 0$*:*

$$\mathsf{Adv}^{\text{omuf}}_{\mathsf{AT},\mathcal{A},\ell}(\lambda) := \Pr[\text{OMUF}_{\mathsf{AT},\mathcal{A},\ell}(\lambda) = 1] = \mathsf{negl}(\lambda),$$

*where* $\text{OMUF}_{\mathsf{AT},\mathcal{A},\ell}$ *is the game defined in Figure 3.*

Next, we want to provide user anonymity. The right notion for this is unlinkability, which guarantees that even colluding issuers and verifiers are unable to link tokens. Arbitrary metadata is a strong way of creating a link, and we omit this problem by only considering fixed public metadata for this notion. Notice that the adversary may query the user oracles for any public metadata $\mathsf{md}$, but that we expect the post-processing to implicitly fail if $\mathsf{md} \neq \mathsf{md}'$. This is in line with for example expiry dates, which would otherwise have been solved in practice using key rotation, and the definition is (as usual) also using a fixed key. Private metadata is outside the control of the user, and gives one bit leakage. We fix it for this game. Note that the adversary controls the keys, and that we therefore do not need to provide access to signing and verification oracles.

**Definition 7 (Unlinkability).** *An anonymous token scheme* $\mathsf{AT}$ *is* $\kappa$*-unlinkable if for any* PPT *adversary* $\mathcal{A}$*, fixed* $b$*,* $\mathsf{md}$*, and any* $m > 0$*,*

$$\mathsf{Adv}^{\text{unlink}}_{\mathsf{AT},\mathcal{A},m,[b],[\mathsf{md}]}(\lambda) := \Pr\big[\text{UNLINK}_{\mathsf{AT},\mathcal{A},m,[b],[\mathsf{md}]}(\lambda) = 1\big] \quad \leq \frac{\kappa}{m} + \mathsf{negl}(\lambda),$$

*where* $\text{UNLINK}_{\mathsf{AT},\mathcal{A},m}$ *is the game defined in Figure 4.*

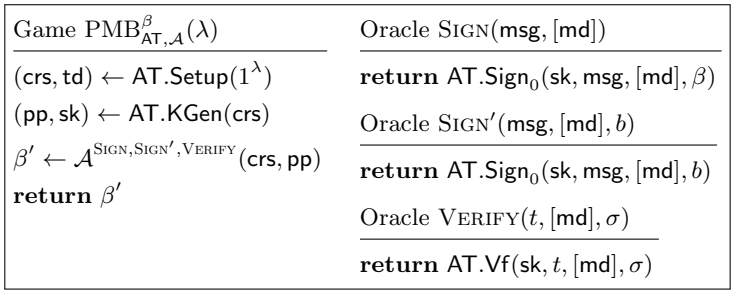


**Fig. 5.** Game for private metadata bit for anonymous tokens.

We finally consider the private metadata bit. We give the adversary access to two signing oracles: One uses the adversary's chosen private bit, the other is using a fixed bit for the game. The adversary can also query a verification oracle. At the end, the adversary outputs its guess for the fixed challenge bit.

**Definition 8 (Private metadata bit).** *An anonymous token scheme* $\mathsf{AT}$ *provides private metadata bit if for any* $\mathsf{PPT}$ *adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathrm{pmb}}_{\mathsf{AT},\mathcal{A}}(\lambda) := \left|\Pr[\mathrm{PMB}^{0}_{\mathsf{AT},\mathcal{A}}(\lambda)] - \Pr[\mathrm{PMB}^{1}_{\mathsf{AT},\mathcal{A}}(\lambda)]\right| = \mathsf{negl}(\lambda),$$

*where* $\mathrm{PMB}^{\beta}_{\mathsf{AT},\mathcal{A}}$ *is the game defined in Figure 5.*

## 4 Anonymous Token Protocols

The Privacy Pass protocol [DGS+18] and its siblings [HIJ+21, KLOR20a] are based on Verifiable Oblivious Pseudo-Random Functions (VOPRF). Here, a user holds some secret input $x$ and the signer holds a secret key $k$ and they evaluate the function $F$ obliviously such that the user learns $F(x,k)$ but nothing about $k$, and the signer learns nothing about the input $x$ nor the output $F(x,k)$. Additionally, the user is ensured that the function is evaluated by the correct secret key.

We give three protocols for Anonymous Tokens (AT) with 1) public metadata, 2) public and private metadata, and 3) public metadata and public verifiability, respectively, constructed from the same framework.

At the core of our protocols lies a verifiable key transformation. Let $d := \mathtt{H}_m(\mathsf{md})$ and the curve point $U := [d]G + K$, where $G$ is a public generator and $K$ is the public key with a corresponding private key $k$. Let $e = (d+k)^{-1}$ be the new signing key and $W' = [e]T'$. Notice the relation

$$\mathsf{KT} : \log_G([d]G + K) = (d+k) = \log_{W'} T'. \tag{1}$$

### 4.1 Secure Key Transformation

We argue that the key-transformation from $k$ to $e$ is secure against one-more unforgeability attacks. Several papers has been written using this transformation. Boneh and Boyen [BB04] shows that this transformation is secure against a non-adaptive attacker for arbitrary metadata $\mathsf{md}$ when used for signatures. Furthermore, Dodis and Yampolskiy [DY05] shows that this transformation is secure against active attackers when the set of possible metadata values is small, and give applications to PRFs. However, these works only prove security with respect to a fixed generators, while our construction signs arbitrary new generators in each execution of the protocol. Recently, Tyagi *et al.* [TCR+21] proved that this transformation is secure against an active attacker with respect to arbitrary generators and arbitrary set of metadata. They reduce the security of the transform to a new one-more gap strong inversion Diffie-Hellman problem (see Section 2.1). They also show that this new problem is equivalent to the simpler $q$-DL assumption. We summarize these results in a lemma.

**Lemma 1.** *Let* $\mathsf{AT}$ *be a scheme with keys* $(\mathsf{pk}, \mathsf{vk})$ *with security property* $P$ *within adversarial advantage* $\mathsf{Adv}^{\mathrm{p}}_{\mathsf{AT},\mathcal{A}}(\lambda)$*, and assume we can prove the relation in Equation 1 within adversarial advantage* $\mathsf{Adv}^{\mathrm{rel}}_{\mathsf{KT},\mathcal{A}}(\lambda)$*. Then* $\mathcal{A}$ *has advantage* $\mathsf{Adv}^{\mathrm{p}}_{\mathsf{AT},\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathrm{rel}}_{\mathsf{KT},\mathcal{A}}(\lambda)$ *against property* $P$ *in the scheme* $\mathsf{AT}$ *with transformed keys* $(\{e = (\mathsf{md} + \mathsf{sk})^{-1}, [e]G\})$.

### 4.2 Anonymous Tokens with Public Metadata

In Figure 6 we present an extension of Privacy Pass [DGS+18] with public metadata. The protocol is designated verifier, as the secret key is needed to verify tokens.

**Setup and Key Generation.** Let $\lambda$ be the security parameter, let $p$ be a prime and let $E$ be an elliptic curve group of order $p$ with generator $G$. Let $\mathtt{H}_t : \{0,1\}^* \to E$ and $\mathtt{H}_m : \{0,1\}^* \to \mathbb{Z}_p$ be hash functions, and assume that group elements and integers can be encoded uniquely as strings. Furthermore, let metadata $\mathsf{md}$ be an element of a public set of valid strings. Finally, let $\mathsf{sk} := k \leftarrow\!\!\$\ \mathbb{Z}_p^*$ be the signing key, and let $\mathsf{pk} := K := [k]G$ be the public key. We consider $G, E, p, \mathtt{H}_t, \mathtt{H}_m$ and $K$ to be implicit knowledge in Figure 6.

**Signing and Verification.** The anonymous tokens protocol in Figure 6 uses the $\Pi_{\mathrm{DLEQ}}$-protocol defined in Section 2.3. The signer computes a proof $\pi_{\mathtt{DLEQ}} := (c, z)$ of equality of discrete logarithms by instantiating the protocol $\Pi_{\mathrm{DLEQ}}(G, T', K, W'; e)$. Given the public parameters $G$ and $K$, and $U := [d]G + K$, this is a proof that $\log_G U = d + k = \log_{W'} T'$. This proves that $W' = [e]T'$, where $e := (d + k)^{-1}$, is computed correctly with respect to $d$ and $K$. To verify, the user instantiates the verification algorithm, denoted by $\mathtt{V}(\pi_{\mathtt{DLEQ}})$.

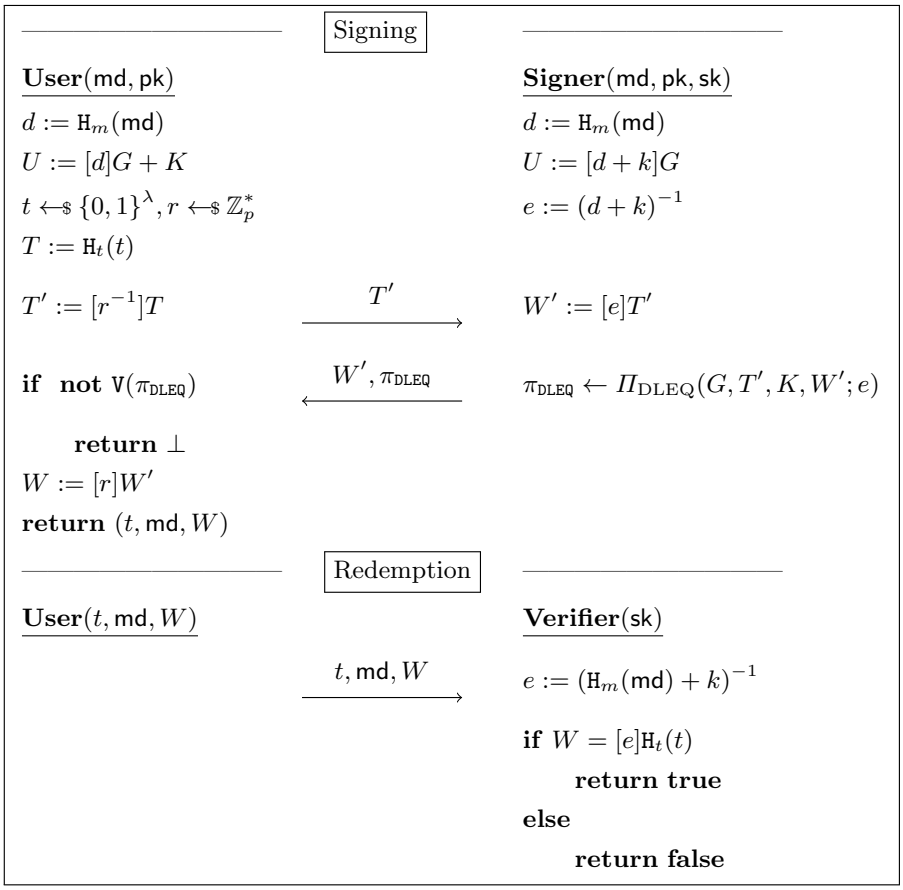


**Fig. 6.** Designated verifier anonymous tokens with public metadata. Our protocol is a direct extension of Privacy Pass [DGS+18].

**Theorem 1 (Completeness).** *The anonymous token protocol with public metadata in Figure 6 is complete according to Definition 4.*

*Proof.* The completeness follows from expanding $W$:

$$W = [r]W' = [r][e]T' = [r][e][r^{-1}]T = [e]\mathtt{H}_t(t).$$

□

**Theorem 2 (Unforgeability).** *The anonymous token protocol with public metadata in Figure 6 achieve one-more unforgeability with respect to Definition 6.*

*Proof.* Using the key transformation as described in Lemma 1, the security of the protocol reduces to the security of the one-more gap strong inversion Diffie-Hellman game as shown in Figure 1. The advantage of an attacker follows from Definition 1 and is proven secure by Tyagi *et al.* [TCR+21, Theorem 1].

□

**Theorem 3 (Unlinkability).** *Fix metadata* md*. Within the set defined by all tokens using* md*, the anonymous token protocol with public metadata in Figure 6 achieve unlinkability with respect to Definition 7.*

*Proof.* This proof is identical to [DGS+18, Theorem 1]: As we sample $r \leftarrow\!\$\ \mathbb{Z}_p$ uniformly at random, it follows that our protocol is unconditionally unlinkable. Since $T$ is a generator of $E$, then $T' = [r^{-1}]T$ is uniformly random and contain no information about $t$ nor $T$. As the signer only sees $T'$, and the verifier only receive $t$, and they are independent, there is no link between the view of the signer and the view of the verifier.

□

### 4.3 AT with Public and Private Metadata

In Figure 7, we present an extension of the PMBTokens [KLOR20a, Figure 8] with public metadata. This protocol is also designated verifier, requiring the secret key for verification.

**Setup and Key Generation.** Let $\lambda$ be the security parameter, let $p$ be a prime and let $E$ be an elliptic curve group of order $p$ with generators $G_0, G_1$. Let $\mathtt{H}_t : \{0,1\}^* \to E$, $\mathtt{H}_m : \{0,1\}^* \to \mathbb{Z}_p^*$ and $\mathtt{H}_s : \{0,1\}^* \to \mathbb{Z}_p^*$ be hash-functions, and assume that group elements and integers can be encoded uniquely as strings. Furthermore, let metadata md be an element of a public set of valid strings. Finally, let $\mathsf{sk} := (k_{0,0}, k_{0,1}, k_{1,0}, k_{1,1}) \leftarrow\!\$\ (\mathbb{Z}_p^*)^4$ (all $k_{i,j}$ being distinct) be the signing key, and let $\mathsf{pk} := \{K_{i,j}\} = \{[k_{i,j}]G_i\}$, for $i, j = 0, 1$, be the public key. This is implicit knowledge in the protocol description.

**Signing and Verification.** The anonymous tokens protocol in Figure 7 uses the $\Pi_{\mathrm{DLEQ2}}$-protocol defined in Section 2.4 twice as a subroutine to ensure that we afterwards can prove that the signed token $W'$ was computed correctly. Given the generators $G_0, G_1, T', S'$, the public keys $K_{i,j} := [k_{i,j}]G_i$ and the elements $V_i := [e_{i,0}]T' + [e_{i,1}]S'$, for $i, j = 0, 1$, we want to prove that the following relations hold:

$$\begin{bmatrix} G_0 + G_1 \\ V_i \end{bmatrix} = [e_{i,0}] \begin{bmatrix} [d]G_0 + K_{i,0} \\ T' \end{bmatrix} + [e_{i,1}] \begin{bmatrix} [d]G_1 + K_{i,1} \\ S' \end{bmatrix}.$$

We instantiate $\Pi_{\mathrm{DLEQ2}}(G_0, G_1, V_0; e_{0,0}, e_{0,1})$ and $\Pi_{\mathrm{DLEQ2}}(G_0, G_1, V_1; e_{1,0}, e_{1,1})$ in Figure 7 to get proofs $\pi_{\mathtt{AND}}$ and $\pi'_{\mathtt{AND}}$. We denote the verification by $\mathtt{V}(\pi_{\mathtt{AND}}, \pi'_{\mathtt{AND}})$.

We also use the $\Pi_{\mathrm{DLEQOR2}}$-protocol defined in Section 2.5. The signer computes an OR-proof of equality of discrete logs by instantiating the zero-knowledge protocol $\Pi_{\mathrm{DLEQOR2}}(G_0, G_1, T', S', V_0, V_1, W'; e_{0,0}, e_{0,1}, e_{1,0}, e_{1,1})$. Consider the generators $G_0, G_1, T', S'$, hashed metadata $d$ and computed value $W'$. The signer then proves that $W'$ is correctly computed, with respect to $T'$ and $S'$, and in

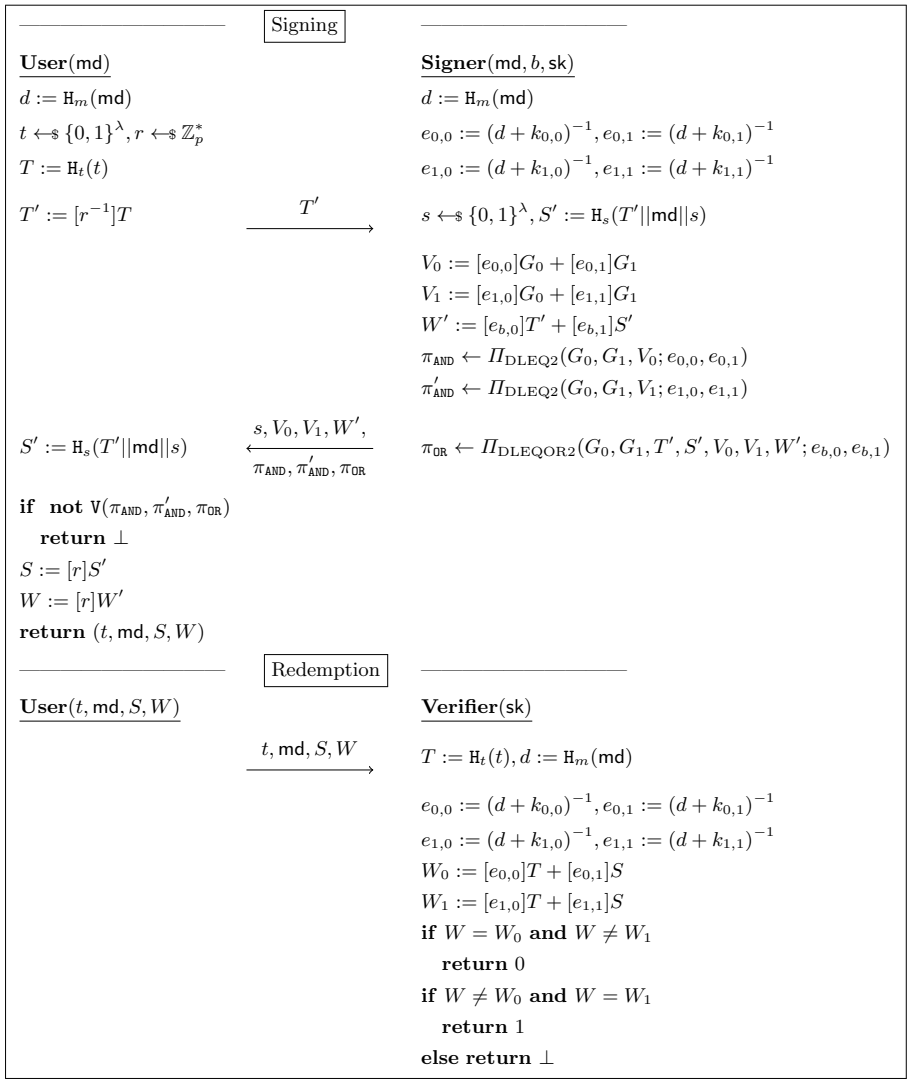


**Fig. 7.** Designated verifier anonymous tokens with public and private metadata, an adjusted extension of Kreuter *et al.* [KLOR20a].

the same way as one of the committed values $V_0$ or $V_1$, with respect to $G$ and $H$. That is, for either $b = 0$ or $b = 1$:

$$V_b = [e_{b,0}]G_0 + [e_{b,1}]G_1 \quad \wedge \quad W' = [e_{b,0}]T' + [e_{b,1}]S'.$$

We denote the verification of the proof $\pi_{\texttt{OR}}$ by $\texttt{V}(\pi_{\texttt{OR}})$.

**Theorem 4 (Completeness).** *The anonymous token protocol with public and private metadata in Figure 7 is complete according to Definition 4 and will, according to Definition 5, return the correct metadata bit except with negligible probability.*

*Proof.* If the user submits $(t, \mathsf{md}, S, W)$, completeness follows from expanding $W_b$:

$$\begin{aligned} W_b &= [r]([e_{b,0}]T' + [e_{b,1}]S') = [r]([e_{b,0}][r^{-1}]T + [e_{b,1}]S') \\ &= [e_{b,0}]T + [r][e_{b,1}]S' = [e_{b,0}]\texttt{H}_t(t||\mathsf{md}) + [e_{b,1}]S. \end{aligned}$$

Furthermore, the probability that this equation holds for both $b = 0$ and $b = 1$ is negligible. If that was the case, then

$$[e_{0,0}]T + [e_{0,1}]S = [e_{1,0}]T + [e_{1,1}]S.$$

As we require all keys $k_{i,j}$ to be distinct, it follows that all $e_{i,j}$ are distinct. Then, we have that

$$T = \left[\frac{e_{1,1} - e_{0,1}}{e_{0,0} - e_{1,0}}\right] S.$$

Since $T = \texttt{H}_t(t)$ is sampled independently and uniformly at random, the probability that this equation holds is $1/p$, which is negligible.

□

**Theorem 5 (Unforgeability).** *The anonymous token protocol with public and private metadata in Figure 7 achieves one-more unforgeability with respect to Definition 6.*

*Proof.* For fixed metadata $\mathsf{md}$ we let the adversary query the signing oracle $\ell$ times for both $b = 0$ and $b = 1$. Using the key transformation as described in Lemma 1, the security of the protocol reduces to the security of the one-more gap strong inversion Diffie-Hellman game as shown in Figure 1. The advantage of an attacker follows from Definition 1 and is proven secure by Tyagi *et al.* [TCR+21, Theorem 1].

□

**Theorem 6 (Unlinkability).** *Fix private metadata $b$ and public metadata $\mathsf{md}$. Within the set defined by all tokens using $b$ and $\mathsf{md}$, the anonymous token protocol with public and private metadata in Figure 7 achieves unlinkability with respect to Definition 7.*

*Proof.* We note that it is easy to create many different anonymity sets to distinguish users based on private metadata being $b = 0$ or $b = 1$, and in combination with different values of public metadata $\mathsf{md}$. We restrict the unlinkability to hold for users within the same anonymity sets based on $b$ and $\mathsf{md}$, both sampled according to the real distribution of private and public metadata. Let $\mathfrak{U}_{b,\mathsf{md}}$ be this set, and select two sessions from $\mathfrak{U}_{b,\mathsf{md}}$. Then it follows directly from [KLOR20a, Theorem 9] that the probability of success of the adversary will be upper bounded by $2/m + \mathsf{negl}(\lambda)$.

□

**Theorem 7 (Private metadata bit).** *The anonymous token protocol with public and private metadata in Figure 7 provides private metadata bit with respect to Definition 8.*

*Proof.* This statement follows directly from the proof of [KLOR20a, Theorem 10], which describes a hybrid argument to prove that instances with private bit 0 are indistinguishable from instances with private bit 1. Notice in particular that the extra `OR`-proofs in our protocol are independent of the private bit $b$, and therefore need no additional simulation.

□

### 4.4 Public Verifiability from Pairings

The authors of Privacy Pass [DGS+18] described an application where the issuer and the recipient of a token would be the same entity, possibly separated by time. For the application we present in Section 6, those two roles are in fact separate, and one should therefore have a scheme that supports public verifiability. It remains an open problem to achieve this without pairings, unless we allow for two rounds of communication [AO00, WSMZ06].

We move on to provide a new variant of a partially blinded signature by Zhang, Safavi-Naini and Susilo [ZSS03]. The protocol allows a user and a signer to generate a signature on a user-private message $m$ and agreed-upon metadata $\mathsf{md}$. Both the issuance protocol and the signature consists of a single curve point.

We show that the idea underlying this scheme can be viewed as a combination of Boneh-Lynn-Shacham signatures [BLS01] and Privacy Pass, inheriting its attractive properties from both.

**Setup and Key Generation.** Let $\lambda$ be the security parameter, let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a pairing, where $G_1$, $G_2$ and $g_T$ are generators for their respective prime $p$ order groups. Furthermore, let $\mathtt{H}_1 : \{0,1\}^* \to \mathbb{G}_1$ and $\mathtt{H}_m : \{0,1\}^* \to \mathbb{Z}_p^*$ be hash functions, and assume that group elements and integers can be encoded uniquely as strings. Also, let $\mathsf{md}$ be an element of a public set of valid metadata strings. Finally, let $\mathsf{sk} := k \leftarrow\!\$\, \mathbb{Z}_p^*$ be the signing key, and let $\mathsf{pk} := K = [k]G_2$ be the public key. This is implicit knowledge in the protocol description.

**Signing and Verification.** Recall that the BLS-scheme signs a message $m$ by hashing it to the group generated by $G_1$ and multiplying it with the secret key $k$; $W := [k]\mathtt{H}_1(m)$. The signature can then be verified by checking that

$$\hat{e}(\mathtt{H}_1(m), K) = \hat{e}(W, G_2).$$

Correctness follows from the linearity of the pairing.

We replace $m$ by a token seed $t$, and use the same trick as earlier to concurrently update the key-pair based on metadata. Then we get the following anonymous token scheme:

**Signing** The user sends $T' := [r^{-1}]\mathtt{H}_1(t)$ to the issuer, who returns $W' := [e]T'$, for $e = (d+k)^{-1}$. The user can verify that the signature is correct by checking $\hat{e}(W', U) = \hat{e}(T', G_2)$, for $U := [d+k]G$, and then storing $(t, \mathsf{md}, W = [r]W')$.

**Verification** The user sends $(t, \mathsf{md}, W)$, and the recipient can verify the token by checking if $\hat{e}(W, U) = \hat{e}(T, G_2)$.

This scheme hides the token similarly to Privacy Pass, it can be verified without using the private key, and its unforgeability follows directly from BLS. We note that the check $\hat{e}(W', U) = \hat{e}(T', G_2)$ ensures that the tokens are signed correctly with respect to the public key. The complete protocol is listed in Figure 8. Finally, we note that we can batch-verify $n$ tokens under the same key and metadata and check for equality by computing

$$\hat{e}\left(\sum_i [c_i]W_i, U\right) = \hat{e}\left(\sum_i [c_i]T_i, G_2\right),$$

where $c_1, \ldots, c_n$ are random coefficients. This saves the verifier of $2(n-1)$ expensive pairing-computations, which is especially useful in systems with large anonymity sets. Note that the verifier computes $T_i$ from the received pre-tokens $t_i$, making sure that $(T_i, W_i)$ is not just a scaling of a different valid token.

**Theorem 8 (Completeness).** *The anonymous token protocol with public metadata and public verifiability in Figure 8 is complete according to Definition 4.*

*Proof.* Completeness follows from expanding $\hat{e}(W, U)$:

$$\begin{aligned}\hat{e}(W, U) &= \hat{e}([r]W', [d+k]G_2) = \hat{e}([r][e]T', [d+k]G_2) \\ &= \hat{e}([r][e][r^{-1}]T, [d+k]G_2) = \hat{e}([e]T, [d+k]G_2) \\ &= \hat{e}(T, G_2)^{e\cdot(d+k)} = \hat{e}(T, G_2).\end{aligned}$$

□

**Theorem 9 (Unforgeability).** *The anonymous token protocol with public metadata and public verifiability in Figure 8 achieve one-more unforgeability with respect to Definition 6.*

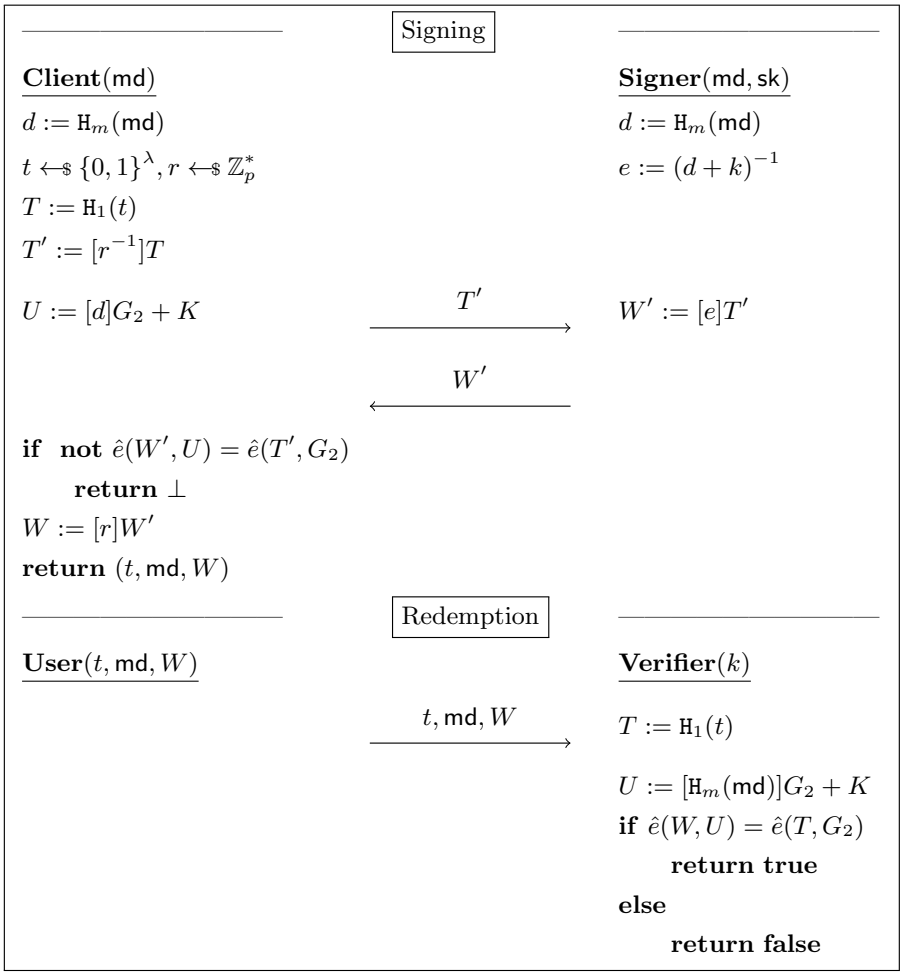


**Fig. 8.** Anonymous tokens with public metadata and public verifiability by adjusting Zhang *et al.* [ZSS03] for asymmetric pairings.

*Proof.* Assume that we have an adversary who breaks unforgeability. In particular, this means that they can produce $\ell+1$ distinct and valid tuples $(t_i, \mathsf{md}, \sigma_i)$ but only query the signing oracle at most $\ell$ times. We use this adversary to construct an adversary against the $(m, n)$-OM-GAP-SDHI problem in the Random Oracle Model.

$\mathcal{A}_1^{\text{om-gap-sdhi}}$ Recall that we assume that the user and the signer agrees on the metadata. Run $\mathcal{O}_m$ on all acceptable metadata values to get the list $\{c_i\}$, and return it.

$\mathcal{A}_2^{\text{om-gap-sdhi}}$ Receive the input $G, K = [k]G, [y_i]G$. Set $\mathsf{vk} = K$ and the other parameters appropriately. Reprogram $\mathcal{O}_1$ such that it on input $t$ returns $[t][y_j]G$ for the next $j$, which looks like a random group element. Whenever the adversary queries the SIGN oracle, forward the query to the SDH oracle. If the adversary wins the OMUF game for some metadata value $\mathsf{md}$ corresponding to an index $\gamma$, we have $\ell+1$ signatures $\sigma_i = [e_{\mathsf{md}}]\mathcal{O}_1(t_i) = [e_{\mathsf{md}}][t_i][y_1]G$. Use the programming of $\mathcal{O}_1$ to return $(\gamma, ([t^{-1}]\sigma_i, \alpha_i))$.

The result from $\mathcal{A}_2^{\text{om-gap-sdhi}}$ satisfies the $(m, n)$-OM-GAP-SDHI conditions.

One can construct a more detailed proof along the lines of [TCR+21, Appendix B] in order to get concrete bounds.

□

**Theorem 10 (Unlinkability).** *Fix metadata* $\mathsf{md}$*. Within the set defined by all tokens using* $\mathsf{md}$*, the anonymous token protocol with public metadata and public verifiability in Figure 8 achieve unlinkability with respect to Definition 7.*

*Proof.* Observe that given any valid token $(t, \mathsf{md}, W)$ and any honestly generated view $(T', W')$ there exists a unique value $r'$ such that both $W - [r']W'$ and $T - [r']T'$ holds, and hence, $T$ is independent of any $W$. It follows that the anonymous token is unlinkable.

□

## 5 Performance and Comparison

In this section, we briefly describe the most efficient anonymous single-use token protocols with public metadata in the literature, for example, to enable batched revocation. We only consider protocols with one round of communication. We compare the protocols with our schemes in Table 1. To streamline the comparison, we assume that all parties know the public metadata, for example that $\mathsf{md}$ is the current date, and assume that this implicit knowledge is not sent. We instantiate the schemes with $\lambda = 128$ bits of security. Finally, we present a concrete example to show that we can replace DIT with our protocol in Figure 6 to improve both communication size and computational efficiency.

### 5.1 Anonymous single-use Tokens with Public Metadata

**Privacy Pass.** Our protocol in Figure 6 is inspired by Privacy Pass [DGS+18], and they have identical structure and communication. The main difference is the change of private key used for signing, and the updated zero-knowledge proof with respect to the new public key, both depending on the public metadata. The zero-knowledge proofs are of the same size, and it follows that the communication sizes are equal. However, Privacy Pass does not allow public metadata unless we have one public key for each valid string of metadata, and hence, to allow for $2^N$ possible messages $\mathsf{md}$, Privacy Pass must publish $2^N$ public keys.

**DIT: De-Identified Authenticated Telemetry at Scale.** DIT [HIJ+21] is also inspired by Privacy Pass [DGS+18], but uses an attribute-based VOPRF to generate new public keys on the fly. To allow for $2^N$ strings of public metadata, there are two main differences: 1) the public key consists of $N+2$ group elements, and 2) the token consists of an additional $N$ group elements and zero-knowledge proofs to ensure that the correct public key is used in the signature.

**Tokens from RSA.** Abe and Fujisaki [AF96] presents a partially blind signature scheme based on RSA. The public exponent $e$ must be at least two bits longer than the public metadata, and we fix this to be of length 130 bits. The user updates the public key to $e_{\mathsf{md}} = e \cdot \tau(\mathsf{md})$, for a public formatting function $\tau$, when they blind the message, and the signer updates the secret key $d_{\mathsf{md}} = (e \cdot \tau(\mathsf{md}))^{-1} \mod N$ when signing. Otherwise, the partially blind signature scheme [AF96] is similar to the blind signature by Chaum [Cha82].

**Tokens with Private Metadata.** Kreuter *et al.* [KLOR20a] presents an extension of Privacy Pass [DGS+18] to include private metadata. They publish two public keys, and the signer proves in zero-knowledge that the token is signed with one of the corresponding private keys. To ensure metadata privacy, each token is randomized based on a fresh seed $s$ that is given to the user, and hence, the signature consists of a seed, a group element, and a proof. The token consists of the initial seed $t$ in addition to two group elements. Like Privacy Pass, this protocol must publish a new pair of public keys for each valid string of metadata.

### 5.2 Comparison

We present a comparison of schemes in Table 1, where we focus on communication complexity. We note that both RSA and pairing based cryptography is usually slower than elliptic curve cryptography, in addition to requiring larger parameters. We also note that the updated keys in our protocols are only dependent on the secret key and the metadata, and can often be pre-computed. We conclude that when allowing for batched token-revocation, our protocols are more efficient than the state of the art in all categories.

While RSA and elliptic curve cryptography are primitives implemented in all mainstream cryptographic libraries, there are few trustworthy implementations

of pairings. Even though there exists a few implementations[7], they are mostly for academic use, maybe except for the implementation in Rust used by Zcash[8]. We refer to [TCR+21, Table 1] for a comparison in computation between some protocols.

| Public Metadata (PM) | PubKey | Request | Signature | Token |
|---|---|---|---|---|
| Privacy Pass [DGS+18] | $257 \cdot 2^N$ | 257 | 769 | 385 |
| DIT [HIJ+21] | $257 \cdot (N+2)$ | 257 | $769 \cdot (N+1)$ | 385 |
| Our scheme (Figure 6) | 257 | 257 | 769 | 385 |
| **PM + Private Metadata** | **PubKey** | **Request** | **Signature** | **Token** |
| Kreuter *et al.* [KLOR20a] | $514 \cdot 2^N$ | 257 | 1921 | 642 |
| Our Scheme (Figure 7) | 1028 | 257 | 3203 | 642 |
| **PM + Public Verifiability** | **PubKey** | **Request** | **Signature** | **Token** |
| Abe and Fujisaki [AF96] | 3202 | 3072 | 3072 | 3200 |
| Our scheme (Figure 8) | 763 | 382 | 382 | 510 |

**Table 1.** Size given in bits. We compare the schemes for 128 bits of security, allowing for $2^N$ strings $\mathsf{md}$ of metadata. Token seed $t$ is of size 128 bits, and metadata $\mathsf{md}$ is implicit knowledge. Privacy Pass, DIT, Kreuter *et al.* and our protocols in Figure 6 and 7 are instantiated with curve x25519 [Ber05], Abe and Fujisaki is instantiated with RSA-3072 and our protocol in Figure 8 is instantiated with BLS12-381 [YCKS21].

### 5.3 Telemetry Collection in WhatsApp

DIT [HIJ+21] was designed to allow users of WhatsApp to anonymously report telemetry data to Facebook. We present a concrete comparison to our protocols in Table 2. Here, we assume that Facebook wants to update their public keys only once a year, rotate signing keys every day, and only sign one token per user each day. We fix a year and encode public metadata as strings "YYYY-MM-DD".

Privacy Pass [DGS+18] is very efficient in terms of communication, but requires one public key per day. Hence, the public key is of size 93805 bits over a year of 365 days, that is, approximately 12 KB. An alternative method to download all keys and store them until usage is to use a Merkle-tree for key-transparency and give paths corresponding to the current public key as a part of each signature. Then, the public key consists of the root of size 256 bits, while each signature consists of $\lceil \log_2(365) \rceil = 9$ hashes of 256 bits in addition to the public key, the token, and the zero-knowledge proof. We give both instantiations in the table, and denote the alternative protocol as Privacy Pass+.

Our scheme in Figure 6 has the smallest overall communication complexity of all schemes. It offers much smaller keys than Privacy Pass, and much smaller signatures than Privacy Pass+ and DIT, saving up to 90 % in communication. If all 2 billion users of WhatsApp report their telemetry every day, our scheme

[7] Pairings: hackmd.io/@zkteam/eccbench
[8] Zcash: github.com/zkcrypto/bls12_381

in Figure 6 would save more than 1.7 TB of communication for the Facebook servers on a daily basis compared to the current implementation of DIT.

Our scheme in Figure 8 offers similar improvements to communication, in addition to public verifiability using pairings, but at the cost of less standardized cryptography and less efficient computation.

| Protocol | PubKey | Request | Signature | Token |
|---|---|---|---|---|
| Privacy Pass [DGS+18] | 93805 | 257 | 769 | 385 |
| Privacy Pass+ | 256 | 257 | 3330 | 385 |
| DIT [HIJ+21] | 2313 | 257 | 7690 | 385 |
| Our scheme (Figure 6) | 257 | 257 | 769 | 385 |
| Our scheme (Figure 8) | 763 | 382 | 382 | 510 |

**Table 2.** Size given in bits. We compare Privacy Pass, DIT, and the protocols in Figure 6 and Figure 8 with daily key-rotation in a year, signing one token at a time.

## 6 Application to Contact Tracing

As nations started adopting digital contact tracing during the COVID-19 pandemic, privacy experts warned that such systems could enable the collection of people's contact graphs. The dp$^3$t protocol [T+20] was eventually adopted as the *de facto* method for digital contact tracing through its implementation and deployment in iOS and Android as the Exposure Notification System (ENS).

We provide a brief overview of the basic dp$^3$t idea in order to put our contribution into context. The protocol is instantiated on each participating phone, which generates a random key (Temporary Exposure Key, TEK) every day. The TEK is used to generate new Rotating Proximity Identifiers (RPI) every 10–20 minutes, which is then broadcast from the phone using Bluetooth Low Energy (BLE). Other phones in the proximity store any RPI they hear.

If Alice tests positive for COVID-19 she can upload her TEKs (now renamed to diagnosis keys, DK) along with her BLE transmission strength to a health authority bulletin board. Bob's phone regularly checks the board to see if there is a sufficiently large overlap between published the DKs and the RPIs stored locally, and with sufficiently low difference between transmission strength and received strength. If this is the case, then Bob is given a suitable alert to let him know that he most likely has been in close vicinity of an infected individual, and should follow any advice given by the health authorities.

The process of uploading TEKs should depend on some sort of authorization. The dp$^3$t documentation describes a simplified model where a doctor receives the test results, and sends the patient an SMS with a short upload code. Now, this process may take precious person-hours during a pandemic. Some countries have therefore opted to connect their exposure notification with already existing centralized registries of positive test results, e.g., Norway, Denmark, and Estonia.

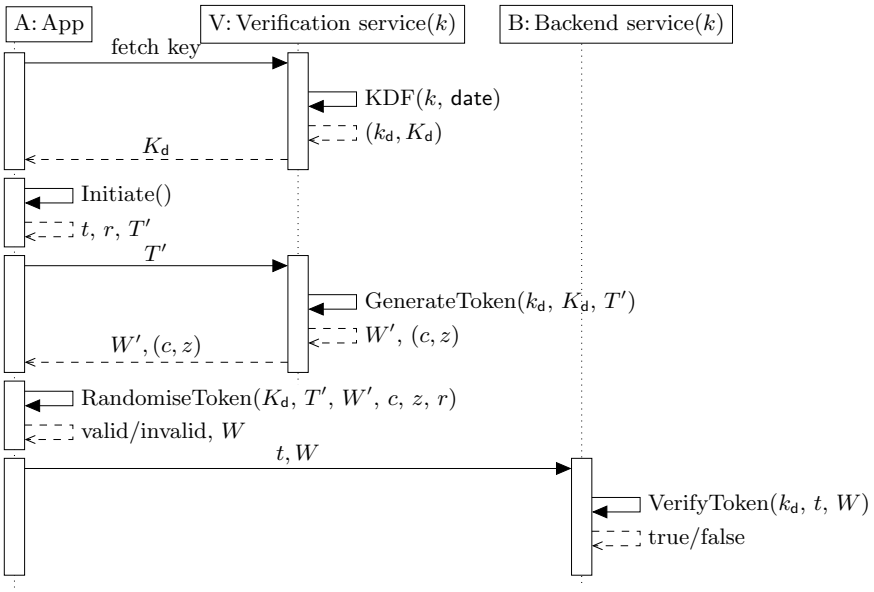


**Fig. 9.** A sequence diagram of anonymous tokens in the Norwegian app Smittestopp.

When starting the upload process, the user is prompted to log in to some government service ("verification"). Once the user has identified herself, the service makes a query to the relevant health registry. The service returns an access token to the app if there exists a recent positive test, which is then used to upload the keys to "backend". Unfortunately, this token may create an identifiable link from the meant-to-be-anonymous database of DKs, and unique identities in the health registry. Using anonymous single-use tokens, one can break this link (up to traffic analysis, e.g., logging timings and network addresses).

The Norwegian Institute of Public Health (NIPH) wanted the tokens to be timestamped in order to avoid users posting severely delayed keys: this would have allowed an attacker to get well again, move back out among other people, and only then upload to the backend service. Notice that merely tying the token to keys – e.g., by using a hash of the TEKs as the token seed $t$ – would not avoid this attack, as those could have been generated and stored until the time of the attack. As a result, it was decided that the keys should be rotated regularly.

The original Privacy Pass protocol was reimplemented as a reusable C# package, to ease the integration into the Norwegian contact tracing app Smittestopp. The verification and backend services keep a master secret key $k$, and generate daily keys from some $\mathsf{KDF}(k, \mathsf{date})$. The public key is posted from the verification service. The full integration of anonymous tokens is described in Figure 9.

We finally note that this key distribution method suffers from a potential attack by a dishonest verification service that could serve special public keys to track individuals. It is, however, detectable by the users if they share their view

of the public keys with each other to ensure consistency. The current solution was accepted by all involved stakeholders due to limited time and a weighting of the practical risk against the potential reward. The challenges with respect to key-rotation and key-sharing strongly motivated the authors' work in Section 4.

## 7 Conclusion

In this work, we have updated the definitions for anonymous single-use tokens to also include public metadata, and we have constructed three protocols that satisfy these definitions. Additionally, we combine public metadata with either private metadata or public verifiability, and show that all instantiations are efficient in practice. For situations with frequent key-rotation, we show that our protocols can save up to 90 % in communication over the state of the art. Furthermore, our protocols fit nicely into the Privacy Pass framework, which makes it easy to incorporate our contributions in the ongoing standardization processes by IETF and W3C, solving an open problem.

We also provide a description of how anonymous one-time tokens can be used to improve the user's privacy in contact tracing applications, and implemented this into the solution used in Norway. The app has more than one million users at the time of writing[9]. As the Norwegian app is built on top of the same code base as the Danish app, we consider it to be easy to extend the adaption of anonymous tokens to their app, and most likely others as well.

We would also like to suggest new use-cases for anonymous tokens. For example, anonymous tokens can improve the privacy of users traveling with public transport. Bus or train companies may require patrons to verify their period tickets for each journey, perhaps primarily to analyze traffic data. However, this can easily reveal the routes of single users while traveling in-between their home and workplace, but also to the abortion clinic, their church or to a public demonstration etc. If all travelers with valid tickets are given a series of tokens (e.g., with public metadata being the date or week or month the ticket is valid), then these can be redeemed when boarding. This way, the companies get the statistics they are interested in, without invading the user's privacy. In general, any systems with leveled authenticated login but anonymous actions can make use of our protocols, e.g., systems with electronic locks that only care if the user has certain privileges or not. We also note that Tyagi *et al.* [TCR+21] detail applications of a construction similar to ours to reduce key management complexity in the OPAQUE password authenticated key exchange protocol, and to ensure stronger security for password breach alerting services.

Finally, we would like to see improvements in three directions. Firstly, the zero-knowledge proofs used by the anonymous tokens protocol with public and private metadata in Figure 7 are much larger than the ones by Kreuter *et al.* [KLOR20a], in contrast to our protocol with public metadata in Figure 6 achieving the exact same communication cost as Privacy Pass [DGS+18]. In particular, we would like to reduce the number of proofs and extra group elements in the protocol in Section 4.3 . Secondly, we would like to provide protocols

[9] Smittestopp: fhi.no/om/smittestopp/nokkeltall-fra-smittestopp, last accessed 2021-12-01.

free of zero-knowledge proofs, to reduce the communication and computational cost, as provided in [KLOR20a, Section 7]. Finally, we would like to extend our protocols to achieve post-quantum security, continuing the work by Albrecht *et al.* [ADDS19] on lattice-based protocols.

**Acknowledgments.** The authors are very grateful to Henrik Walker Moe (Bekk Consulting AS) for stellar collaboration during the C# implementation phase. The final integration into Smittestopp was primarily a collaboration between Henrik, Johannes Brodwall (Sopra Steria) and Sindre Møgster Braaten (NIPH), with the authors and others as close consultants. We thank Nirvan Tyagi, Sofia Celi, Thomas Ristenpart and Christopher Wood for pointing out a flaw in the security game and unforgeability proof in an earlier version of this paper. The second author is grateful to the students Teodor Dahl Knutsen and Tallak Mannum for many useful comments to an earlier version of the manuscript. We would also like thank the anonymous reviewers at PETS 2021 and Financial Crypto 2022 for their feedback which greatly improved the presentation of this paper.

## References

ADDS19. Martin R. Albrecht, Alex Davidson, Amit Deo, and Nigel P. Smart. Round-optimal verifiable oblivious pseudorandom functions from ideal lattices. Cryptology ePrint Archive, Report 2019/1271, 2019. https://eprint.iacr.org/2019/1271.

AF96. Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology – ASIACRYPT'96*, volume 1163 of *Lecture Notes in Computer Science*, pages 244–251. Springer, Heidelberg, November 1996.

AMO08. Norio Akagi, Yoshifumi Manabe, and Tatsuaki Okamoto. An efficient anonymous credential system. In Gene Tsudik, editor, *FC 2008: 12th International Conference on Financial Cryptography and Data Security*, volume 5143 of *Lecture Notes in Computer Science*, pages 272–286. Springer, Heidelberg, January 2008.

AO00. Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 271–286. Springer, Heidelberg, August 2000.

BB04. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, Heidelberg, May 2004.

Ber05. Daniel J. Bernstein. Curve25519: high-speed elliptic curve cryptography, 2005. https://cr.yp.to/ecdh.html.

BL13. Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013: 20th Conference on Computer and Communications Security*, pages 1087–1098. ACM Press, November 2013.

BLS01. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, Heidelberg, December 2001.

BLS03. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Constructing elliptic curves with prescribed embedding degrees. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02: 3rd International Conference on Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 257–267. Springer, Heidelberg, September 2003.

BMR+17. Jonathan Burns, Daniel Moore, Katrina Ray, Ryan Speers, and Brian Vohaska. EC-OPRF: Oblivious pseudorandom functions using elliptic curves. Cryptology ePrint Archive, Report 2017/111, 2017. http://eprint.iacr.org/2017/111.

BNPS02. Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme. In Paul F. Syverson, editor, *FC 2001: 5th International Conference on Financial Cryptography*, volume 2339 of *Lecture Notes in Computer Science*, pages 319–338. Springer, Heidelberg, February 2002.

Bol03. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, Heidelberg, January 2003.

BPV12. Olivier Blazy, David Pointcheval, and Damien Vergnaud. Compact round-optimal partially-blind signatures. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12: 8th International Conference on Security in Communication Networks*, volume 7485 of *Lecture Notes in Computer Science*, pages 95–112. Springer, Heidelberg, September 2012.

CG08. Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 2008: 15th Conference on Computer and Communications Security*, pages 345–356. ACM Press, October 2008.

Cha82. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology – CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982.

Cha83. David Chaum. Blind signature system. In David Chaum, editor, *Advances in Cryptology – CRYPTO'83*, page 153. Plenum Press, New York, USA, 1983.

CHL05. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 302–321. Springer, Heidelberg, May 2005.

CHYC05. Sherman S. M. Chow, Lucas Chi Kwong Hui, Siu-Ming Yiu, and K. P. Chow. Two improved partially blind signature schemes from bilinear pairings. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP 05: 10th Australasian Conference on Information Security and Privacy*, volume 3574 of *Lecture Notes in Computer Science*, pages 316–328. Springer, Heidelberg, July 2005.

CKS09. Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In Stanislaw Jarecki and Gene Tsudik, editors, *PKC 2009: 12th Interna-*

*tional Conference on Theory and Practice of Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 481–500. Springer, Heidelberg, March 2009.

CL01. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, Heidelberg, May 2001.

CL03. Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02: 3rd International Conference on Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer, Heidelberg, September 2003.

CL04. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer, Heidelberg, August 2004.

CMZ14. Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. Algebraic MACs and keyed-verification anonymous credentials. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 2014: 21st Conference on Computer and Communications Security*, pages 1205–1216. ACM Press, November 2014.

CP93. David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer, Heidelberg, August 1993.

CPZ20. Melissa Chase, Trevor Perrin, and Greg Zaverucha. The signal private group system and anonymous credentials supporting efficient verifiable encryption. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 20: 27th Conference on Computer and Communications Security*, pages 1445–1459. ACM Press, November 2020.

CV02. Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In Vijayalakshmi Atluri, editor, *ACM CCS 2002: 9th Conference on Computer and Communications Security*, pages 21–30. ACM Press, November 2002.

CZMS06. Xiaofeng Chen, Fangguo Zhang, Yi Mu, and Willy Susilo. Efficient provably secure restrictive partially blind signatures from bilinear pairings. In Giovanni Di Crescenzo and Avi Rubin, editors, *FC 2006: 10th International Conference on Financial Cryptography and Data Security*, volume 4107 of *Lecture Notes in Computer Science*, pages 251–265. Springer, Heidelberg, February / March 2006.

Dav21. Alex Davidson. Supporting the latest version of the privacy pass protocol. https://blog.cloudflare.com/supporting-the-latest-version-of-the-privacy-pass-protocol, 2021. (Accessed 01-December-2021).

DGS$^+$. Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: A privacy-enhancing protocol and browser extension. https://privacypass.github.io. (Accessed 01-December-2021).

DGS$^+$18. Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: Bypassing internet challenges anonymously. *Proceedings on Privacy Enhancing Technologies*, 2018(3):164–180, July 2018.

DY05. Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 416–431. Springer, Heidelberg, January 2005.

FHKS16. Georg Fuchsbauer, Christian Hanser, Chethan Kamath, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model from weaker assumptions. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16: 10th International Conference on Security in Communication Networks*, volume 9841 of *Lecture Notes in Computer Science*, pages 391–408. Springer, Heidelberg, August / September 2016.

FHS15. Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 233–253. Springer, Heidelberg, August 2015.

FIPR05. Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In Joe Kilian, editor, *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 303–324. Springer, Heidelberg, February 2005.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, Heidelberg, August 1987.

GMS10. Jorge Guajardo, Bart Mennink, and Berry Schoenmakers. Anonymous credential schemes with encrypted attributes. In Swee-Huay Heng, Rebecca N. Wright, and Bok-Min Goi, editors, *CANS 10: 9th International Conference on Cryptology and Network Security*, volume 6467 of *Lecture Notes in Computer Science*, pages 314–333. Springer, Heidelberg, December 2010.

Hen14. Ryan Henry. *Efficient Zero-Knowledge Proofs and Applications*. PhD thesis, University of Waterloo, 2014.

HG13. Ryan Henry and Ian Goldberg. Batch proofs of partial knowledge. In Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS 13: 11th International Conference on Applied Cryptography and Network Security*, volume 7954 of *Lecture Notes in Computer Science*, pages 502–517. Springer, Heidelberg, June 2013.

HIJ+21. Sharon Huang, Subodh Iyengar, Sundar Jeyaraman, Shiv Kushwah, Chen-Kuei Lee, Zutian Luo, Payman Mohassel, Ananth Raghunathan, Shaahid Shaikh, Yen-Chieh Sung, and Albert Zhang. Dit: De-identified authenticated telemetry at scale. Technical report, Facebook Inc., https://research.fb.com/wp-content/uploads/2021/04/DIT-De-Identified-Authenticated-Telemetry-at-Scale_final.pdf, 2021.

HS21. Lucjan Hanzlik and Daniel Slamanig. With a little help from my friends: Constructing practical anonymous credentials. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21. Association for Computing Machinery, 2021.

Int21. Internet Engineering Task Force. Privacy pass datatracker. https://datatracker.ietf.org/wg/privacypass, 2021. (Accessed 01-Dec-2021).

IT21. Subodh Iyengar and Erik Taubeneck. Fraud resistant, privacy preserving reporting using blind signatures. https://github.com/siyengar/private-fraud-prevention, 2021. (Accessed 01-December-2021).

JKK14. Stanislaw Jarecki, Aggelos Kiayias, and Hugo Krawczyk. Round-optimal password-protected secret sharing and T-PAKE in the password-only model. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 233–253. Springer, Heidelberg, December 2014.

JKX18. Stanislaw Jarecki, Hugo Krawczyk, and Jiayu Xu. OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 456–486. Springer, Heidelberg, April / May 2018.

KLOR20a. Ben Kreuter, Tancrède Lepoint, Michele Orrù, and Mariana Raykova. Anonymous tokens with private metadata bit. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 308–336. Springer, Heidelberg, August 2020.

KLOR20b. Ben Kreuter, Tancrede Lepoint, Michele Orru, and Mariana Raykova. Efficient anonymous tokens with private metadata bit. Cryptology ePrint Archive, Report 2020/072, 2020. https://eprint.iacr.org/2020/072.

Oka93. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer, Heidelberg, August 1993.

PS96. David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology – ASIACRYPT'96*, volume 1163 of *Lecture Notes in Computer Science*, pages 252–265. Springer, Heidelberg, November 1996.

PWH+17. Dimitrios Papadopoulos, Duane Wessels, Shumon Huque, Moni Naor, Jan Včelák, Leonid Reyzin, and Sharon Goldberg. Making NSEC5 practical for DNSSEC. Cryptology ePrint Archive, Report 2017/099, 2017. http://eprint.iacr.org/2017/099.

PZ13. Christian Paquin and Greg Zaverucha. U-prove cryptographic specification v1.1 revision 3, 2013. https://www.microsoft.com/en-us/research/project/u-prove.

T+20. Carmela Troncoso et al. Decentralized privacy-preserving proximity tracing. https://arxiv.org/abs/2005.12273, 2020.

TCR+21. Nirvan Tyagi, Sofia Celi, Thomas Ristenpart, Nick Sullivan, Stefano Tessaro, and Christopher A. Wood. A fast and simple partially oblivious prf, with applications. Cryptology ePrint Archive, Report 2021/864, 2021.

TPY+19. Kurt Thomas, Jennifer Pullman, Kevin Yeo, Ananth Raghunathan, Patrick Gage Kelley, Luca Invernizzi, Borbala Benko, Tadek Pietraszek, Sarvar Patel, Dan Boneh, and Elie Bursztein. Protecting accounts from credential stuffing with password breach alerting. In Nadia Heninger and Patrick Traynor, editors, *USENIX Security 2019: 28th USENIX Security Symposium*, pages 1556–1571. USENIX Association, August 2019.

Wor21. World Wide Web Consortium. Trust Token API Explainer. https://github.com/WICG/trust-token-api, 2021. (Accessed 01-December-2021).

WSMZ06. Qianhong Wu, Willy Susilo, Yi Mu, and Fanguo Zhang. Efficient partially blind signatures with provable security. In Marina Gavrilova, Osvaldo Gervasi, Vipin Kumar, C. J. Kenneth Tan, David Taniar, Antonio Laganá, Youngsong Mun, and Hyunseung Choo, editors, *Computational Science and Its Applications - ICCSA 2006*, pages 345–354, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

YCKS21. S. Yonezawa, S. Chikara, T. Kobayashi, and T. Saito. Pairing-Friendly Curves. https://tools.ietf.org/id/draft-yonezawa-pairing-friendly-curves-02.html, 2021. (Accessed 01-December-2021).

ZSS03. Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In Thomas Johansson and Subhamoy Maitra, editors, *Progress in Cryptology - INDOCRYPT 2003: 4th International Conference in Cryptology in India*, volume 2904 of *Lecture Notes in Computer Science*, pages 191–204. Springer, Heidelberg, December 2003.