Post-Quantum Anonymous Tokens

Sébastien Hauri

School of Computer and Communication Sciences

Diploma work

January 2024

Responsible
Prof. Serge Vaudenay
EPFL / LASEC

Supervisor
Laurane Marco
EPFL / LASEC

Contents

1 Introduction

In the complex mosaic of the modern internet infrastructure, Content Delivery Networks (CDNs) are proving to be essential facilitators, streamlining the global distribution of internet content. With centralised control over content distribution, CDNs act as global arbiters, wielding the authority to allow or block content requests in a bid to thwart malicious traffic. However, this centralised control encounters a significant challenge with the advent of privacy tools such as Tor and VPNs, which contribute to the emergence of shared IP addresses. This shared context poses a problem, as it becomes increasingly challenging for CDNs to distinguish between legitimate users and potential threats. The response to this challenge has often taken the form of CAPTCHAs, an acronym for Completely Automated Public Turing test to tell Computers and Humans Apart. While effective in curbing malicious activities and distinguishing between human users and automated bots, CAPTCHAs introduce a different kind of problem. Users frequently encounter inconvenience and annoyance, forced to decipher distorted characters or solve puzzles to access desired content. This trade-off between security and user experience highlights the need for innovative solutions that transcend the limitations of conventional CAPTCHA mechanisms. In this landscape, Cloudflare[1], a leading player in the CDN domain, proposed a solution named Privacy Pass [DGS+18]. This novel approach represents a paradigm shift in solving the problems posed by CAPTCHAs, offering a streamlined and privacy-focused method for meeting the challenges of the internet. Privacy Pass introduces the concept of anonymous tokens, which serve as cryptographic passports, enabling users to anonymously authenticate themselves without divulging sensitive information. The high-level design of Privacy Pass revolves around the issuance and validation of these anonymous tokens. During legitimate interactions, users, authenticated through the issuance of tokens, can subsequently present these tokens to Cloudflare-protected websites without undergoing repetitive CAPTCHA challenges. This not only speeds up access for users, but also preserves their privacy by eliminating the need for recurring identification processes.

From there on, several publications emerged with the goal of implementing additional extensions into the protocol. Anonymous tokens with private metadata bit [KLOR20, CDV23] develop a way of transmitting two trust signals, without the user being able to distinguish which signal is embedded into a token. Another desirable functionality is adding public metadata to anonymous tokens. Public metadata facilitates context-aware interactions without revealing sensitive information. It also allows for more efficient key-rotation. [SS22] gave the first construction of such a scheme, with also the ability of combining both public and private metadata. Finally, public verifiability of anonymous tokens was introduced in [SS22, BLOR22]. The primary goal is about introducing a layer of transparency and accountability to the token verification process. In order to provide anonymity, those constructions rely on different primitives such as verifiable oblivious pseudo-random functions (VOPRF), algebraic message authentication codes (MACs) and pairing-based cryptography. Those primitives rely mostly on classical, as in non post-quantum, cryptographic mechanisms.

On another side, the impending arrival of quantum computers announces a transformative era in the realm of cryptography and privacy. Traditional cryptographic algorithms, such as factoring-based and discrete logarithm-based cryptography, which underpin the security of today's communication systems, are broken by quantum computers leveraging Shor's [Sho94] and/or Grover's [Gro96] algorithm. This paradigm shift poses profound implications for data privacy, as confidential information previously considered secure could be exposed. Consequently, the cryptographic community is actively engaged in developing quantum-resistant algorithms to strengthen digital systems against the cryptographic vulnerabilities posed by quantum computing. Currently, the National Institute of Standards and Technology (NIST) of the United States

[1] https://www.cloudflare.com/

has started a process to standardise[2] one or more post-quantum public-key cryptography protocols. On the same topic, some research started emerging on post-quantum anonymous credentials [BLNS23]. Anonymous credentials and anonymous tokens are strongly related, in the sense that the former can be seen as a generalisation of the latter. In [PWFHW23], Cloudflare proposed a post-quantum Privacy Pass through the help of post-quantum anonymous credentials. The protocol also allows for per-client rate-limits, leveraging the power of attributes engraved in the credentials. Although their innovative protocol achieves great performance results, it allows for the retrieval of only 1 credential at a time. Also, we note that their implementation does not allow proofs to be computed with zero-knowledge.

1.1 Contributions

In this work, we first formalise anonymous tokens and their security properties, *unlinkability* and *one-more token unforgeability*. We then summarise the different classical constructions, recalling their underlying primitives and discussing their advantages and inconveniences. We then explore the post-quantum anonymous token schemes and explain how we can build new constructions from other post-quantum primitives. We compare and analyse the different resulting metrics. We then propose a new post-quantum anonymous tokens scheme MQAT, based on the hardness of solving multivariate quadratic equations systems. Our anonymous tokens scheme is publicly verifiable. We prove that under some assumptions, this new construction fulfills the security properties previously defined, and discuss the intuition behind the post-quantum resistance. Finally, we present a concrete instantiation of our new protocol, written in Go. The source code is available online at `https://github.com/sebhauri/mqat`. At the moment, the token issuance protocol takes a bit less than a second and the verification process is performed in the half of it, which makes the construction competitive with other state-of-the-art schemes.

2 Preliminaries

2.1 Notations

Throughout this work, we will use different notations that we explain below. $\mathbb{G}$ is an additive cyclic group, with generator G and prime order p. Elements of the group are usually denoted with capital letters; lower case letters usually denote scalars in the group $\mathbb{Z}_p$. For an integer $n \in \mathbb{N}$, we denote with $[n]$ the set $\{0, \ldots, n-1\}$ of n elements. We use $:=$ for assignments and $\leftarrow$ for outputs of probabilistic algorithms. When sampling uniformly at random an element x from the set $\mathcal{S}$ we write $x \leftarrow\!\!\$\ \mathcal{S}$. We also abuse the notation by writing $x_0, \ldots, x_{n-1} \leftarrow\!\!\$\ \mathcal{S}$ to express the fact that we sample n elements $x_i, i \in [n]$, from the set $\mathcal{S}$ uniformly at random. String concatenation is denoted by $||$. A function $\mu : \mathbb{N} \to [0, 1]$ is said to be negligible (denoted $\mu = \mathsf{negl}(\lambda)$) if for all $c \in \mathbb{N}$ there exists $\lambda_c \in \mathbb{N}$ such that $\mu(\lambda) \leq \lambda^{-c}$ for all $\lambda \geq \lambda_c$. We call *advantage* the probability that an adversary $\mathcal{A}$ wins a security game Game, and it is denoted as $\mathsf{Adv}^{\mathsf{Game}}_{\mathcal{A},Params}(\lambda) = \Pr[\mathsf{Game}^{\mathcal{A}}_{Params}(\lambda) \to 1]$.

2.2 Anonymous tokens

We describe below the interface for anonymous tokens, its correctness definition as well as the different security notions. An anonymous token scheme is composed of two phases: 1) the issuance phase, where a client (also called user) and a server (also called issuer) get involved in an interactive protocol and the client ends up with a (possibly several) token(s), and 2) the redeem phase, where a client *spends* a token it has previously been issued and a verifier checks its validity. There is no restriction on an issuer also being a verifier, but note that this is often the case.

[2] `https://csrc.nist.gov/projects/post-quantum-cryptography`

Definition 1 (Anonymous tokens)**.** *An* anonymous tokens scheme AT *consists of the following algorithms:*

- $\mathsf{AT.Setup}(1^\lambda) \to \mathsf{cpp}$*, the setup algorithm that takes as input the security parameter* λ *(in unary form) and outputs the common public parameters* cpp*. All the remaining algorithms are assumed to have the common public parameters as their first input and we will thus sometimes omit it.*
- $\mathsf{AT.KeyGen}(\mathsf{cpp}) \to (\mathsf{pk}, \mathsf{sk})$*, the key generation algorithm that takes as input the common public parameters* cpp *and outputs the public key* pk *and the secret key (also sometimes called signing key)* sk*, both belonging to the* issuer*.*
- $\langle \mathsf{AT.User}(\mathsf{pk}), \mathsf{AT.Sign}(\mathsf{sk}) \rangle \to \mathsf{t}$*, the interactive token issuance protocol between the* user *and the* issuer*. The* $\mathsf{User}(\cdot)$ *algorithm takes as input the* issuer's *public key* pk *and the* $\mathsf{Sign}(\cdot)$ *algorithm takes as input the* issuer's *secret key* sk*. At the end of the protocol, the* issuer *outputs nothing and the* user *outputs a tag* t *and a signature* σ *or* $\perp$ *if it failed. The pair* $\mathsf{t} := (t, \sigma)$ *is called a token. This algorithm is a 2-move interactive protocol, initiated by the client, and can be defined by the three following algorithms:*
  - $\mathsf{AT.User}_0(\mathsf{pk}) \to (\mathsf{st}, \mathsf{query})$
  - $\mathsf{AT.Sign}_0(\mathsf{sk}, \mathsf{query}) \to \mathsf{resp}$
  - $\mathsf{AT.User}_1(\mathsf{st}, \mathsf{pk}, \mathsf{resp}) \to (t, \sigma)$
- $\mathsf{AT.Verify}(\mathsf{sk}, \mathsf{t}) \to \mathsf{bool}$*, the verification algorithm that takes as input the* issuer's *secret key* sk*, a token* t *and outputs a boolean value indicating if the token is valid or not.*

An anonymous token scheme AT is *correct* if for any honestly generated token and honest verification the token verifies. In other words, let λ be the security parameter, $\mathsf{cpp} \leftarrow \mathsf{AT.Setup}(1^\lambda)$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{AT.KeyGen}(\mathsf{cpp})$ then AT is *correct* if

$$\Pr[\mathsf{AT.Verify}(\mathsf{sk}, \langle \mathsf{AT.User}(\mathsf{pk}), \mathsf{AT.Sign}(\mathsf{sk}) \rangle) = 1] = 1 - \mathsf{negl}(\lambda) \ .$$

Unlinkability ensures that the token issuance and the token redemption are *unlinkable*, meaning that when a token is redeemed the verifier cannot link it to an issuing session better than randomly guessing on those. Formally, the κ*-unlinkability* property states that if ℓ tokens were issued but not yet redeemed, an adversary can not link the issuance session of a token when seeing the remaining $\ell - 1$ tokens in a random order with probability better than $\frac{\kappa}{\ell}$.

Definition 2 (κ-unlinkability)**.** *Let* UNLINK *be the security game defined in Figure 1, we say that an anonymous token scheme* AT *is* κ-unlinkable *if for any probabilistic polynomial-time (PPT) adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ *and any* $\ell > 0$*:*

$$\mathsf{Adv}^{\mathsf{UNLINK}}_{\mathcal{A},\ell}(\lambda) \leq \frac{\kappa}{\ell} + \mathsf{negl}(\lambda)$$

Unforgeability ensures that a cheating user cannot trick a verifier into accepting more tokens than it has issued to that specific user. Put in other words, a user should not be able to forge a valid token from the issuer. We call this property the *one-more token unforgeability.*

Definition 3 (One-more unforgeability)**.** *Let* OMUF *be the security game defined in Figure 2, we say that an anonymous token scheme* AT *is* one-more unforgeable *if for any PPT adversary* $\mathcal{A}$ *and any* $\ell \geq 0$*:*

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathcal{A},\ell}(\lambda) = \mathsf{negl}(\lambda)$$

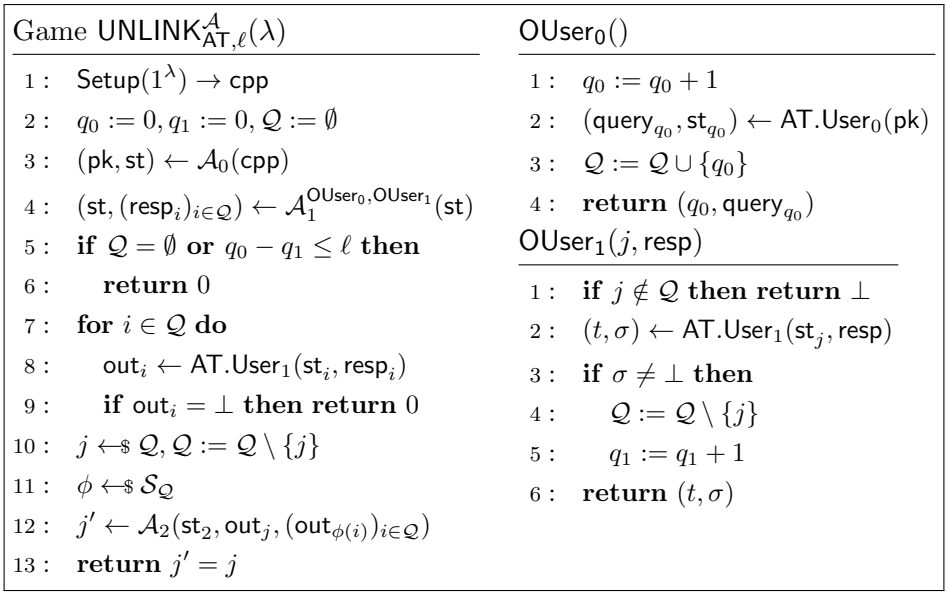

Figure 1: Unlinkability *security game for an anonymous token scheme* AT.

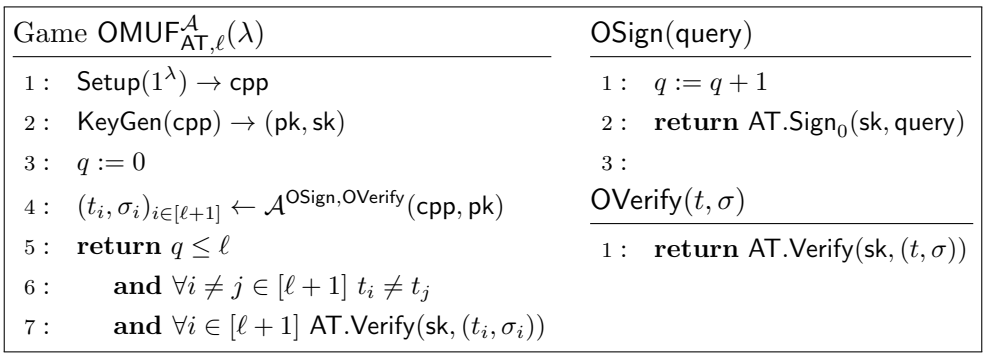

Figure 2: One-more unforgeability *security game for an anonymous token scheme* AT.

In the next sections, we will present several classical anonymous tokens constructions and discuss their security properties as well as their limitations. Anonymous tokens have been initially proposed in Privacy Pass [DGS+18] that we present first. This development led to other constructions of anonymous tokens implementing additional extensions, such as having private/public metadata embedded into tokens or public verifiability of tokens, that we present after. Throughout the different schemes, we simplify the token issuance to one token at a time and stress out that this does not reduce the security of the protocols. Also, we do not completely state the token redemption phase, but rather explain how the server checks that a token is valid.

2.2.1 Privacy Pass

The Privacy Pass protocol has been developed by Cloudflare in [DGS+18]. It is a pioneering practical solution of anonymous tokens, which is based on a verifiable oblivious pseudo-random function (VOPRF). An OPRF allows two parties, generally a client and a server, to compute together a function $F(k; x) = z$ (which is often shortened as $F_k(x)$), with x and k being respectively the client and the server private inputs. Either party should not be able to learn any information about the private input of the other. We say that an OPRF is *verifiable* when the client can be convinced that the server did the computation correctly. To prove the latter, the server will often send back the computation along with a non-interactive zero-knowledge (NIZK)

proof of the computation. As an example, let $T \in \mathbb{G}$ and $k \in \mathbb{Z}_p$ be the private input of the client and the server respectively. Then the protocol presented in Figure 3 is an OPRF that evaluates $F_k(T) = k \cdot T$ since $W = r^{-1} \cdot W' = r^{-1}k \cdot T' = r^{-1}kr \cdot T = k \cdot T$.

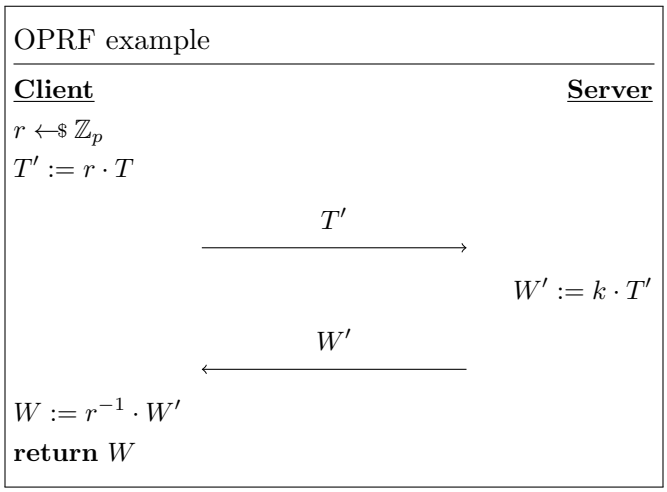

Figure 3: *Example* OPRF $F_k(T) = k \cdot T$.

The scheme. We present below the Privacy Pass protocol between a client and a server. Let $k \in \mathbb{Z}_p$ be the issuer's private key with the corresponding public key $K = k \cdot G \in \mathbb{G}$ which the issuer committed to[3], $H_t(t) : \{0,1\}^\lambda \to \mathbb{G}$ an hash function and Π be a NIZK proof system of discrete log equality. The issuance protocol is described in Figure 4. The user first samples a

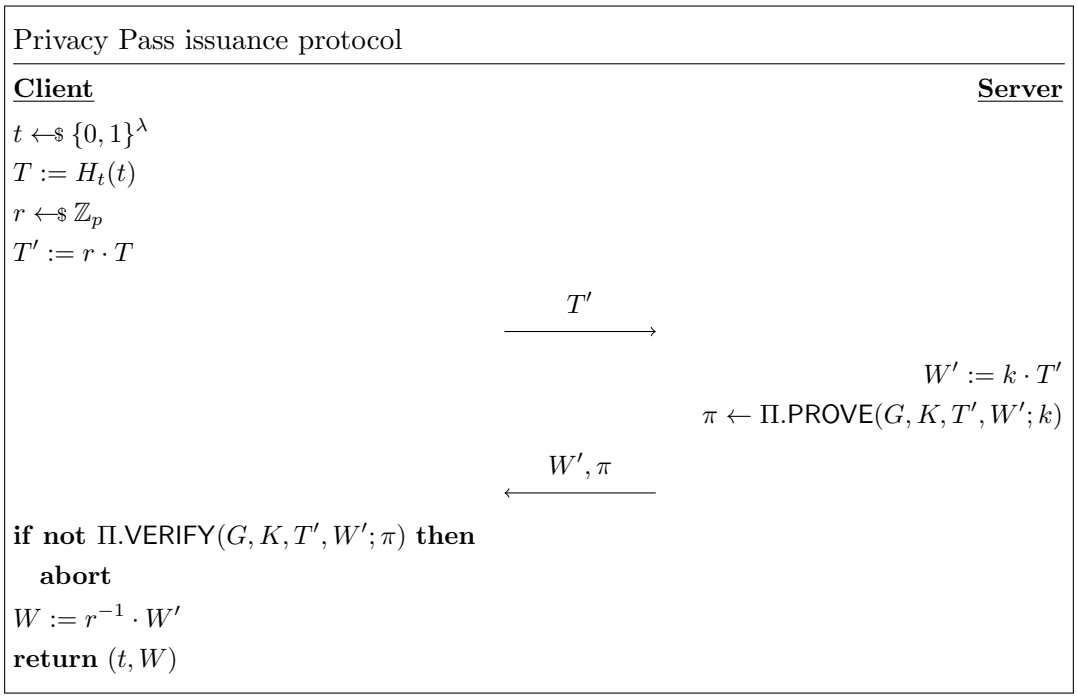

Figure 4: *The interactive token issuance protocol from* Privacy Pass.

random tag t that it hashes into $T \in \mathbb{G}$. The client blinds this value T with a random value

[3]This prevents the server from tracking users by signing tokens with a per-user key. From now on, we assume that each issuer in the following schemes committed to their private key.

r, called the blinding factor, and sends this masked value to the issuer. The latter signs it, computes a NIZK proof that it did the computation correctly and sends back the signature and the proof to the user. Finally, the user checks the proof, unblinds the signature to obtain a signature W on T, and stores the token $\mathtt{t} = (t, W)$. As we can see, the Privacy Pass protocol is based on the VOPRF $F_k(t) = k \cdot H_t(t)$. On receiving the token (t, W), the verifier computes $T := H_t(t)$ and checks that $W = k \cdot T$. This anonymous token protocol is correct from the correctness properties of both the NIZK proof and the VOPRF.

Security. The protocol proposed in [DGS+18] fulfills the two anonymous tokens security properties, *unlinkability* and *one-more token unforgeability*. The former is ensured by the help of the blinding factor r and the VOPRF. Intuitively, as r is chosen uniformly at random in $\mathbb{Z}_p$ by the client, T' is uniformly distributed in $\mathbb{G}$. Using this and the fact that the VOPRF blinds the user input correctly, the server cannot link the issuance and redemption phases. Regarding the latter, without going into further details, the *one-more token unforgeability* game related to this construction can be reduced to the security of the El-Gamal encryption scheme. The scheme is then 1-*unforgeable*.

2.2.2 Anonymous tokens with private metadata bit

[KLOR20] came with the idea of including a private metadata bit into the token. The principle is the following: if the clients have a good reputation the server will give them a valid token and otherwise an invalid one. Recall that anonymous tokens had as a first goal to prove the trustworthiness of internet requests without compromising on the user's privacy. One issue with this solution was that if an issuer stopped providing malicious users with tokens, they would know that they had been spotted. One could then train a machine learning model to detect which malicious behaviour goes unnoticed and use this knowledge to still get tokens when acting malicious. The goal of those new constructions is to encode this *"validity"* into the private metadata bit, and the user should not be able to learn whether it has received an invalid token before it redeems the latter. This property is called the *privacy of the metadata bit*. It has been chosen to restrict the private metadata to only one bit to ensure that it does not lead to de-anonymization of tokens.

The PMBT scheme. The naive way of encoding a private metadata bit into a token is to use the Privacy Pass protocol from [DGS+18] described above and encode the bit with the secret key of the server: one key for valid tokens and a second one for invalid tokens. This construction does not fulfill the *privacy of the metadata bit* property since the underlying primitive, the VOPRF, is completely deterministic: when requesting a signed token, the user could always ask for the same token and will know that it is invalid when the token it receives changes. On the other hand, the server cannot tell that the user always asks for the same token, as the request is blinded (recall the blinding factor r). [KLOR20] came with a construction called Private Metadata Bit Tokens (PMBT). The high-level idea is the following: generalise the Privacy Pass protocol to allow for randomised tokens and use two different keys for each bit respectively. We describe below the details of the protocol.

Let H be another generator of $\mathbb{G}$, $((x_0, x_1), (y_0, y_1)) \in \mathbb{Z}_p^2 \times \mathbb{Z}_p^2$ be the issuer's private keys with the corresponding public keys $X_0 = x_0 \cdot G + y_0 \cdot H$ and $X_1 = x_1 \cdot G + y_1 \cdot H$ such that $X_0 \neq X_1$, b be the private bit of the server, H_t, H_s be two random oracles $\{0,1\}^* \to \mathbb{G}$ and Π be a non-interactive zero-knowledge proof system of an OR discrete log equality. We present the PMBT construction in Figure 5. On receiving the token (t, S, W), the server computes $T := H_t(t)$ and reads the private metadata bit by checking either if $W = x_0 \cdot T + y_0 \cdot S$ or $W = x_1 \cdot T + y_1 \cdot S$.

PMBT issuance protocol

Client		**Server**
$t \leftarrow\$ \{0,1\}^\lambda$		
$r \leftarrow\$ \mathbb{Z}_q$		
$T := H_t(t)$		
$T' := r^{-1} \cdot T$		
	$\xrightarrow{T'}$	
		$s \leftarrow\$ \{0,1\}^\lambda$
		$S' := H_s(T', s)$
		$W' := x_b \cdot T' + y_b \cdot S'$
		$\pi \leftarrow \Pi.\mathsf{PROVE}(G, H, X_0, X_1, T', S', W'; x_b, y_b)$
	$\xleftarrow{s, W', \pi}$	
$S' := H_s(T', s)$		
if not $\Pi.\mathsf{VERIFY}(G, H, X_0, X_1, T', S', W'; \pi)$ **then**		
abort		
$S := r \cdot S'$		
$W := r \cdot W'$		
return (t, S, W)		

Figure 5: *The issuance protocol of* PMBT.

Unfortunately, [KLOR20] does not directly provide a verification algorithm. Also, there is a problem with this construction, that is, if an adversary has access to a verification oracle, then given two tokens (t_0, S_0, W_0) and (t_1, S_1, W_1), if $t_0 = t_1$ then by setting $t^* := t_0 = t_1$, $S^* := 2S_0 - S_1$ and $W^* := 2W_0 - W_1$ the token (t^*, S^*, W^*) is valid only if the same data bit was used to generate t_0 and t_1 and the adversary can use this property to win the *privacy of the private metadata bit* security game. They proposed a counter-measure by stating a verification algorithm that is secure for this kind of attack.

The ATHM scheme. [CDV23] noted the problem discussed above and tried to find a way to mitigate the issue. To do so, they decided to change the PMBT protocol in several ways:

- Change the underlying VOPRF by an algebraic message authentication code (MAC) in order to get rid of the former's deterministic property.
- Change the fact that the randomness of the token is only chosen by the user.

Algebraic MACs have been introduced by [DKPW12] and then generalised in [CMZ14] to provide a keyed-verification anonymous credentials (KVAC) scheme. Unlike other MAC constructions that rely on pseudo-random functions, algebraic MACs rely on specific number theory properties to provide the same level of security. We briefly recall hereafter one of the two proposed schemes of [CMZ14], $\mathsf{MAC}_{\mathsf{GGM}}$. In what follows, let $\mathbf{m} = (m_1, ..., m_n)$ be a list of n messages in a field $\mathbb{F}_p$ of prime order p. $\mathsf{MAC}_{\mathsf{GGM}}$ is defined in the following way:

- The secret key $\mathbf{x}$ is chosen randomly in $\mathbb{F}_p^{n+1}$
- The tag σ is computed as follows: chose $U \leftarrow\$ \mathbb{G} \setminus \{0\}$, compute $U' := (x_0 + \sum_{i=1}^{n} x_i m_i) \cdot U$ and output $\sigma := (U, U') \in \mathbb{G}^2$

- The tag σ on message $\mathbf{m}$ is verified as follows: parse $(U, U') := \sigma$, accept if $U \neq 0$ and $U' = (x_0 + \sum_{i=1}^{n} x_i m_i) \cdot U$

We will present only the Anonymous Tokens with Hidden Metadata bit (ATHM) construction based on $\mathsf{MAC}_{\mathsf{GGM}}$, but a quite similar construction with $\mathsf{MAC}_{\mathsf{DDH}}$, the other algebraic MAC proposed in [CMZ14], can easily be adapted from it. Let $(x, y, z) \in \mathbb{Z}_p \times (\mathbb{Z}_p^*)^2$ be the private key of the server and $Z = z \cdot G$ its corresponding public key, b be its private input and Π be a simulatable non-interactive proof system. The ATHM issuance protocol is presented in Figure 6. When a token (t, P, Q) is redeemed, the server verifies that $P \neq 0$, checks which $b \in \{0, 1\}$ fulfills the equality $Q = (x + by + tz) \cdot P$ and either returns the bit or aborts if there is none.

ATHM issuance protocol

Client		**Server**
$t_C \leftarrow\$ \mathbb{Z}_p$		
$r \leftarrow\$ \mathbb{Z}_p$		
$T := t_C \cdot Z + r \cdot G$		
	$\xrightarrow{T}$	
		$t_S \leftarrow\$ \mathbb{Z}_p$
		$d \leftarrow\$ \mathbb{Z}_p^*$
		$U := d \cdot G$
		$V := d(x \cdot G + by \cdot G + t_S z \cdot G + T)$
		$\pi \leftarrow \Pi.\mathsf{PROVE}(b, d, x, y, z; U, V, t_S, T, Z)$
	$\xleftarrow{U, V, t_S, \pi}$	
if not $\Pi.\mathsf{VERIFY}(\pi, U, V, t_S, T, Z)$ **then**		
abort		
if $U = 0$ **then abort**		
$c \leftarrow\$ \mathbb{Z}_p^*$		
$P := c \cdot U$		
$Q := c(V - r \cdot U)$		
$t = t_C + t_S$		
return (t, P, Q)		

Figure 6: *The issuance protocol of* ATHM.

Regarding the security of the scheme, the *unlinkability* property is ensured thanks to the blinding factors r and c, and also the verification of Π that checks that b is in fact just one bit. The *one-more token unforgeability* is ensured by the security of $\mathsf{MAC}_{\mathsf{GGM}}$ against forgeries. In fact, the authors of [CMZ14] proved that $\mathsf{MAC}_{\mathsf{GGM}}$ was *existentially unforgeable under chosen message attacks* given a verification oracle in the generic group model. One should also note that the server needs to pay attention to the double spending of a token, as the token $(t, \ell \cdot P, \ell \cdot Q)$ is also valid for any $\ell \in \mathbb{Z}_p^*$. The *privacy of the metadata bit* property is ensured by the issuer always sampling a new uniformly random value t_S. Otherwise, an attacker reusing t_C could do the same attack as before and learn if the bit was the same.

2.2.3 Anonymous Tokens with Public Metadata

Another type of anonymous tokens constructions are the ones that add public metadata to tokens. A variant of ATHM with public metadata can be found in Appendix B1 of [CDV23]. There are also schemes that seek to find public verifiability. Those constructions could be really helpful in settings where the verifier is different from the issuer. In such cases, the verification algorithm from Definition 1 should be changed and take as input the issuer's public key. [SS22] came with those exact ideas that we present below.

ATPM. The first idea consists of including public metadata into existing anonymous token constructions such as Privacy Pass [DGS+18] and PMBT [KLOR20]. [SS22] achieves this by applying a key transformation on the the signing keys of the issuer: instead of using the VOPRF $F_k(t) = k \cdot H_t(t)$, they use the VOPRF $F_e(t) = e \cdot H_t(t)$ with $e = d + k$ where $d = H_m(\mathsf{md})$ is the hash of the public metadata md. We do not present those constructions as they only add this subtlety to the protocols.

Public verifiability. As the second idea also relies on the key transformation presented above, this construction develops an other paradigm: issuing tokens that any entity could verify given the public key of the issuer. This new scheme is based on pairings and more specifically short signatures introduced by [BLS01]. Pairing-based cryptography uses a pairing between two cryptographic groups (often chosen as elliptic curve groups) mapping to a third one by the help of a function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ which satisfies two properties:

1. bilinearity: $e(aP, bQ) = ab \cdot e(P, Q)$
2. non-degeneracy: $e \neq 1$

Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a pairing with G_1, G_2, G_T being a generator of their respective groups of prime order p, let $H_t : \{0,1\}^* \to \mathbb{G}_1$ and $H_m : \{0,1\}^* \to \mathbb{Z}_p^*$ be two hash functions, $k \in \mathbb{Z}_p^*$ the issuer's private key with $K = k \cdot G_2 \in \mathbb{G}_2$ its corresponding public key and finally md be an element of a public set of valid metadata strings. The issuing protocol for anonymous tokens with public metadata and public verifiability is presented in Figure 7. Upon receiving the token (t, md, W) the verifier first computes $T := H_t(t)$ and $U := H_m(\mathsf{md}) \cdot G_2 + K$ and then outputs the result of the check $\hat{e}(W, U) = \hat{e}(T, G_2)$. The protocol fulfills both *one-more token unforgeability* and *unlinkability*.

In this section, we proposed an interface for anonymous tokens and defined the two related security properties *unlinkability* and *one-more token unforgeability*. We have seen that anonymous tokens can be built on top of classical cryptographic primitives such as VOPRFs or algebraic MACs and also saw that more functionalities can be added into tokens, such as private/public metadata or public verifiability.

The issuance protocol with public metadata and public verifiability

Client		**Server**
$d := H_m(\mathsf{md})$		$d := H_m(\mathsf{md})$
$t \leftarrow\$ \{0,1\}^\lambda$		$e := (d+k)^{-1}$
$r \leftarrow\$ \mathbb{Z}_p^*$		
$T := H_t(t)$		
$T' := r^{-1} \cdot T$		
	$\xrightarrow{T'}$	
		$W' := e \cdot T'$
	$\xleftarrow{W'}$	
$U := d \cdot G_2 + K$		
if not $\hat{e}(W', U) = \hat{e}(T', G_2)$ **then**		
abort		
$W := r \cdot W'$		
return (t, md, W)		

Figure 7: *Issuing protocol for anonymous tokens with public verifiability.*

3 Towards post-quantum anonymous tokens

In the following sections, we will briefly overview several post-quantum primitives that could be useful for building secure post-quantum anonymous tokens schemes. We will then compare the different existing post-quantum anonymous tokens constructions with some new constructions that could be built on top of other post-quantum primitives. What differs a lot from classical constructions is the latency that can occur due to heavy computations of post-quantum schemes in use. Another big difference is the signature size, which can make a big difference when sending it through high-latency networks. We try to analyse the different advantages and disadvantages of each scheme, and which trade-off can be made to get the better option for post-quantum anonymous tokens.

3.1 Different post-quantum primitives

3.1.1 Lattice-based problems

Lattice-based cryptography stands as a cornerstone in the ongoing quest to fortify digital security against the impending threat posed by quantum computing. As traditional cryptographic schemes face vulnerability to quantum algorithms, lattice-based cryptography provides a promising alternative rooted in the complexity of lattice problems. We present below the most common used problems in lattice-based cryptography. Let $\mathcal{R} = \mathbb{Z}_q[X]/(X^n - 1)$ be the polynomial ring corresponding to the set of polynomials of degree less than n with coefficients in $\mathbb{Z}_q$, q being a prime number or a prime power.

M-SIS. Let d be the rank of the module $M \in \mathcal{R}^d$, $\beta \ll q$ and $m \in \mathbb{N}$. Given $A \leftarrow\$ \mathcal{R}_q^{d \times m}$, find $\mathbf{u} \in \mathcal{R}^m$ such that $||\mathbf{u}|| \leq \beta$ and $A\mathbf{u} = \mathbf{0} \mod q$.

M-LWE. Let d be the rank of the module $M \in \mathcal{R}^d$ and $m \in \mathbb{N}$, let $A \leftarrow\$ \mathcal{R}_q^{m \times d}$, $\mathbf{s} \leftarrow\$ \mathcal{R}^d$ and $\mathbf{e} \leftarrow\$ \mathcal{R}^m$, and set $\mathbf{b} := A\mathbf{s} + \mathbf{e} \mod q$. The goal of the M-LWE problem is to distinguish the

pair $(A, \mathbf{b})$ from a uniformly random pair chosen in $\mathcal{R}_q^{m \times d} \times \mathcal{R}_q^m$.

mat-NTRU. Given integers m, p, q, β with $\gcd(p, q) = 1$, the goal of the mat-NTRU problem is to distinguish between a random matrix $A \leftarrow\$ \mathbb{Z}_q^{m \times m}$ and $B := p^{-1}G^{-1}F \mod q$, for some $F \leftarrow\$ \{0, \pm 1, ..., \pm\beta\}^{n \times n}$ and some $G \leftarrow\$ \{0, \pm 1, ..., \pm\beta\}^{n \times n} \cap (\mathbb{Z}_q^{n \times n})^*$.

These lattice problems are believed to be hard because no polynomial-time algorithms are known for solving them in general. The hardness of lattice problems forms the basis for the security of lattice-based cryptographic schemes, which are being explored as alternatives to traditional cryptographic systems like RSA and ECC (Elliptic Curve Cryptography).

3.1.2 CRYSTALS Dilithium

CRYSTALS Dilithium [DKL+18] is a post-quantum digital signature scheme based on the hardness of the M-LWE problem. Let $(\mathbf{s_1}, \mathbf{s_2}) \in \mathcal{R}^\ell \times \mathcal{R}^k$ be the Dilithium private key with $(A, \mathbf{t})$ the corresponding public key such that $\mathbf{t} = A\mathbf{s_1} + \mathbf{s_2}$. Let $\mathbf{v}$ be a vector over $\mathcal{R}$, the functions HighBits($\mathbf{v}$) and LowBits($\mathbf{v}$) decompose $\mathbf{v}$ uniquely as $\mathbf{v} = \text{HighBits}(\mathbf{v}) \cdot 2\gamma_2 + \text{LowBits}(\mathbf{v})$ such that $-\gamma_2 < \text{LowBits}(\mathbf{v}) \leq \gamma_2$ for some γ_2. The (simplified) Dilithium authentication scheme is defined by the following procedure:

1. The client samples a random $\mathbf{y} \leftarrow\$ \mathcal{R}^\ell$ and sends $\mathbf{w_1} = \text{HighBits}(\mathbf{y})$.
2. The verifier samples $c \leftarrow\$ \mathcal{R}$ and sends it back.
3. The client sends $\mathbf{z} = \mathbf{y} + c \cdot \mathbf{s_1}$.
4. The verifier compute $\mathbf{w_1'} = \text{HighBits}(A\mathbf{z} - c\mathbf{t})$ and accepts if $\mathbf{w_1} = \mathbf{w_1'}$.

By making the previous process non-interactive with $c = H(A||\mathbf{t}||M||\mathbf{w_1})$ for some hash function H and message M, we obtain what is called a Dilithium signature.

3.1.3 The ISIS problem ([BLNS23])

The Inhomogenous Shortest Integer Solution is a new lattice-based assumption defined over the following problem: given a random matrix $A \in \mathbb{Z}_p^{m \times n}$, a function $f : [N] \to \mathbb{Z}_p^n$ and an access to an oracle who chooses a random $x \in [N]$ and outputs it with a small vector $\mathbf{s}$ such that $A\mathbf{s} = f(x)$, find another tuple $(x', \mathbf{s}')$ such that $\mathbf{s}'$ is small and $A\mathbf{s}' = f(x')$. This lattice-based problem allows to create a blind signature scheme: we sign a message x with the pre-image $\mathbf{s}$ if $f(\cdot)$ is modelled as a random oracle. Also, one can give very efficient zero-knowledge proofs of the ISIS_f problem.

3.1.4 Isogenies

Isogeny-based cryptography was introduced by Couveignes [Cou06], Rostostev and Stolbunov [RS06]. It relies on the problem of computing an isogeny between two given elliptic curves. An isogeny is a surjective morphism between two elliptic curves. For cryptographic applications we consider separable isogenies only, which are fully determined by their kernels. Several variants of this foundational problem appear in the literature and have led to a wide range of cryptographic constructions. We distinguish two general line of works, one based on SIDH [JDF11] and its variants (this includes the line of work that offers countermeasures to the recent SIDH attacks [CD23, MMP+23, Rob23]), and the other one based on CSIDH [CLM+18] and similar schemes. They both led to the design of a variety of cryptographic primitives, which we will explore later on in the context of VOPRFs (see Section 3.3.1). The main appeal of isogeny-based cryptography resides in the compactness of its primitives which are usually order of magnitudes smaller than other post-quantum primitives, at the cost of being significantly less efficient.

3.1.5 Multivariate equations systems

Multivariate cryptography has gained a lot of interest in post-quantum cryptography due to its potential resistance to quantum attacks, efficient implementation on constrained devices, and its suitability for post-quantum cryptographic scenarios. It is based on the hardness of solving the $\mathcal{MQ}$ problem and its variants. We recall hereafter the basics of multivariate cryptography.

Definition 4 (The $\mathcal{MQ}$ function [SSH11])**.** *We denote by $\mathcal{MQ}(n, m, \mathbb{F}_q)$ a family of functions*

$$\left\{ \mathcal{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \;\middle|\; \begin{array}{c} f_\ell(\mathbf{x}) = \sum_{i,j} \alpha_{\ell,i,j} x_i x_j + \sum_i \beta_{\ell,i} x_i, \\ \alpha_{\ell,i,j}, \beta_{\ell,i} \in \mathbb{F}_q, \ \ell = 1, \ldots, m \end{array} \right\}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ and constant terms have been omitted for simplicity without any security loss.

With such a family of functions, we can define the $\mathcal{MQ}$ problem as the following:

Definition 5 ($\mathcal{MQ}$ problem)**.** *Given $\mathcal{F} \in \mathcal{MQ}(n, m, \mathbb{F}_q)$, find a vector $\mathbf{x}$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{0}$.*

This problem is proven to be NP-hard and thus its interest in post-quantum cryptography has grown. It is also believed that solving random instances of the $\mathcal{MQ}$ problem is hard. We define below the notion of intractability for the $\mathcal{MQ}$ function ([SSH11] Definition 1).

Definition 6 (Intractability)**.** *For polynomially bounded functions $n = n(\lambda)$, $m = m(\lambda)$ and $q = q(\lambda)$, $\mathcal{MQ}(n, m, \mathbb{F}_q)$ is* intractable *if there is no PPT algorithm that takes as input $(\mathcal{F}, \mathbf{v})$, with $\mathcal{F} \leftarrow\$ \mathcal{MQ}(n, m, \mathbb{F}_q)$, $\mathbf{s} \leftarrow\$ \mathbb{F}_q^n$ and $\mathbf{v} := \mathcal{F}(\mathbf{s})$, and outputs $\mathbf{s}' \in \mathbb{F}_q^n$ such that $\mathcal{F}(\mathbf{s}') = \mathbf{v}$ with non-negligible probability.*

Several public key cryptosystems (PKC) have been built on variants of this problem, and we present some of them below.

Oil and vinegar. The Oil and Vinegar (OV) digital signature scheme has first been introduced in [Pat97]. The principle is simple and explained in this paragraph. A multivariate quadratic equation system can be seen as a map $\mathcal{F} = (f_1(\mathbf{x}), \ldots, f_o(\mathbf{x})) : \mathbb{F}_q^n \to \mathbb{F}_q^o \in \mathcal{MQ}(n, o, \mathbb{F}_q)$. Let $n = o + v$ and define the index sets $V = \{0, \ldots, v-1\}$ and $O = \{v, \ldots, n-1\}$ referred to as the Vinegar set and the Oil set respectively, then each polynomial f_ℓ can be written as

$$f_\ell(\mathbf{x}) = \sum_{j,k \in O} \alpha_{j,k}^{(\ell)} x_j x_k + \sum_{j,k \in V} \beta_{j,k}^{(\ell)} x_j x_k + \sum_{j \in O, k \in V} \gamma_{j,k}^{(\ell)} x_j x_k + \mathbf{L}_\ell(\mathbf{x})$$

where $\mathbf{L}_\ell(\mathbf{x})$ contains the linear coefficients of f_ℓ. By carefully choosing the polynomials such that the first term of the equation always vanishes, then the map $\mathcal{F}$ is easily invertible: we fix the v variables $x_i, i \in V$, and solve a system of o linear equations in o variables $x_i, i \in O$, by using Gaussian elimination. Such polynomials are called OV polynomials and $\mathcal{F}$ an OV map. With this in mind, one can define a digital signature scheme. The signer generates an OV map $\mathcal{F}$ (also called the central map) and a random invertible linear map $\mathcal{T} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ which it uses as the private key. The public key is the map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^o = \mathcal{F} \circ \mathcal{T}$. To get a signature on a message $\mathbf{w} \in \mathbb{F}_q^o$, the signer computes a preimage $\mathbf{y} \in \mathbb{F}_q^n$ of $\mathbf{w}$ under the central map $\mathcal{F}$ and outputs the signature $\mathbf{z} := \mathcal{T}^{-1}(\mathbf{y})$. To check a signature $\mathbf{z} \in \mathbb{F}_q^n$ on a message $\mathbf{w} \in \mathbb{F}_q^o$, the verifier computes $\mathbf{w}' := \mathcal{P}(\mathbf{z})$ and checks that $\mathbf{w} = \mathbf{w}'$. The balanced Oil and Vinegar ($o = v$) digital signature scheme has been broken by the Kipnis-Shamir attack in [KS98]. An unbalanced ($v > o$) version has been proposed in [KPG99]. It is believed that the Unbalanced Oil and Vinegar (UOV) digital signature scheme has the best compromise between security and efficiency when $v = 2o$, which means a number of variables three time as large as the number of equations. A modern version of UOV[4] has been submitted to the NIST additional call for post-quantum digital signature schemes standardisation process[5].

[4] https://www.uovsig.org/
[5] https://csrc.nist.gov/projects/pqc-dig-sig

Rainbow. The Rainbow digital signature scheme has been proposed in [DS05]. It can be seen as a multi-layer version of UOV. The private key is composed of an easily invertible quadratic map $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and two invertible linear maps $\mathcal{S} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $\mathcal{T} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ and the public key is the quadratic map $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}_q^n \to \mathbb{F}_q^m$. To sign a message $\mathbf{w} \in \mathbb{F}_q^m$, one computes recursively $\mathbf{x} := \mathcal{S}^{-1}(\mathbf{w})$, $\mathbf{y} := \mathcal{F}^{-1}(\mathbf{x})$ and $\mathbf{z} := \mathcal{T}^{-1}(\mathbf{y})$. The signature is then the vector $\mathbf{z} \in \mathbb{F}_q^n$. To check the authenticity of a signature $\mathbf{z}$ on message $\mathbf{w}$, one computes $\mathbf{w}' := \mathcal{P}(\mathbf{z})$ and checks if $\mathbf{w} = \mathbf{w}'$. Rainbow was one of the 3 finalists to the NIST call for post-quantum digital signature standardisation process but unfortunately the protocol has lately been broken by [Beu22], in a attack where the secret key can be recovered after on average 53 hours using a standard laptop.

A 5-pass identification scheme. [SSH11] proposed two identification schemes based on the hardness of solving the $\mathcal{MQ}$ problem; a 3-pass and a 5-pass scheme. We quickly recall hereafter the 5-pass identification scheme and its security notions. The scheme is based on the following principle: for any quadratic map $\mathcal{F} \in \mathcal{MQ}(n, m, \mathbb{F}_q)$ its polar form is defined as

$$\mathcal{G}(\mathbf{x}, \mathbf{y}) = \mathcal{F}(\mathbf{x} + \mathbf{y}) - \mathcal{F}(\mathbf{x}) - \mathcal{F}(\mathbf{y}).$$

$\mathcal{G}$ is a bilinear function. The idea is to prove the knowledge of a preimage $\mathbf{s} \in \mathbb{F}_q^n$ (the secret key) of the vector $\mathbf{v} := \mathcal{F}(\mathbf{s}) \in \mathbb{F}_q^m$ (the public key) under the multivariate quadratic map $\mathcal{F} \in \mathcal{MQ}(n, m, \mathbb{F}_q)$, in a zero-knowledge way. To do so, the prover splits the secret key as $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1$, the public key as $\mathcal{F}(\mathbf{s}) = \mathcal{F}(\mathbf{r}_0 + \mathbf{r}_1) = \mathcal{F}(\mathbf{r}_0) + \mathcal{F}(\mathbf{r}_1) + \mathcal{G}(\mathbf{r}_0, \mathbf{r}_1)$ and proves that it knows a tuple $(\mathbf{r}_0, \mathbf{r}_1, \mathbf{t}_0, \mathbf{t}_1, \mathbf{e}_0, \mathbf{e}_1)$ such that

$$(\mathbf{t}_0, \mathbf{e}_0) = (\alpha\mathbf{r}_0 - \mathbf{t}_1, \alpha\mathcal{F}(\mathbf{r}_0) - \mathbf{e}_1) \tag{1}$$

$$\mathcal{G}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0 = \alpha(\mathbf{v} - \mathcal{F}(\mathbf{r}_1)) - \mathcal{G}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{e}_1 \tag{2}$$

where $\alpha \in \mathbb{F}_q$ is chosen by the verifier, $\alpha\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{t}_1$ and $\alpha\mathcal{F}(\mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$. The protocol goes as follows: the prover first samples $\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0$ in their respective domains; it then computes $\mathbf{r}_1 := \mathbf{s} - \mathbf{r}_0$ and makes two commitments $c_0 := Com(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$ and $c_1 := Com(\mathbf{r}_1, \mathcal{G}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0)$, with a commitment scheme Com, that it sends to the verifier; the latter responds with a random value $\alpha \leftarrow\$ \mathbb{F}_q$ with which the prover computes values $\mathbf{t}_1$ and $\mathbf{e}_1$ that it sends back to the verifier; finally the verifier queries the prover by picking a challenge $b \leftarrow\$ \{0, 1\}$ to which the prover responds with $\mathbf{r}_b$; if $b = 0$ the verifier checks that $c_0 = Com(\mathbf{r}_0, \alpha\mathbf{r}_0 - \mathbf{t}_1, \alpha\mathcal{F}(\mathbf{r}_0) - \mathbf{e}_1)$ and otherwise that $c_1 = Com(\mathbf{r}_1, \alpha(\mathbf{v} - \mathcal{F}(\mathbf{r}_1) - \mathcal{G}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{e}_1)$. [SSH11] proved two properties as theorems that we state hereafter without giving the proofs.

Theorem 1. *The 5-pass protocol is statistically zero-knowledge when the commitment scheme Com is statistically hiding.*

Theorem 2. *The 5-pass protocol is an argument of knowledge for the relation* $R_{\mathcal{F}} = \{(\mathbf{v}, \mathbf{x}) \in \mathbb{F}_q^m \times \mathbb{F}_q^n : \mathbf{v} = \mathcal{F}(\mathbf{x})\}$ *with knowledge error* $\frac{1}{2} + \frac{1}{2q}$ *when the commitment scheme Com is computationally binding.*

MQDSS. The MQDSS digital signature protocol has been proposed in [CHR+16]. It is based on a non-interactive version of the 5-pass identification system described above. We outline below how the scheme works. The protocol assumes the existence of an hash function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$ and two pseudo-random number generators $\mathsf{H}_1 : \{0,1\}^{2\lambda} \to \mathbb{F}_q^r$ and $\mathsf{H}_2 : \{0,1\}^{2\lambda} \to \{0,1\}^r$.

Key generation: The secret key consists of a vector $\mathbf{s} \in \mathbb{F}_q^n$ and the public key of a multivariate quadratic system $\mathcal{F} \in \mathcal{MQ}(n, m, \mathbb{F}_q)$ and the vector $\mathbf{v} = \mathcal{F}(\mathbf{s}) \in \mathbb{F}_q^m$.

Signature generation: To sign a message $m \in \{0,1\}^*$, the signer first computes the message dependent values $R := \mathsf{H}(\mathbf{s}||m)$ and $D := \mathsf{H}(R||m)$. It then performs r rounds of the non-interactive version of the 5-pass identification scheme described in [SSH11]. To sample the values $\alpha_0, \ldots, \alpha_{r-1}$ the signer uses the pseudo-random number generator H_1 with input $h_1 = (D, \sigma_0)$ where $\sigma_0 = (c_0^{(0)}||c_1^{(0)}||c_0^{(1)}||c_1^{(1)}||\ldots||c_0^{(r-1)}||c_1^{(r-1)})$ is the concatenation of the $2r$ commitments needed for the r rounds of the 5-pass identification scheme. To sample the values $b_0, \ldots, b_{r-1}$ the signer uses the pseudo-random number generator H_2 with input $h_2 = (h_1, \sigma_1)$ with $\sigma_1 = (\mathbf{t}_1^{(0)}||\mathbf{e}_1^{(0)}||\ldots||\mathbf{t}_1^{(r-1)}||\mathbf{e}_1^{(r-1)})$, the concatenation of the r prover's second messages in the 5-pass identification scheme. The signature is the tuple $\sigma = (R, \sigma_0, \sigma_1, \sigma_2)$ where $\sigma_2 = (\mathbf{r}_{b_0}^{(0)}, \ldots, \mathbf{r}_{b_{r-1}}^{(r-1)})$.

Signature verification: To check that a signature σ is valid for message m, the verifier first computes $D = \mathsf{H}(R||m)$. It then composes h_1 and h_2 to derive the challenges $\alpha_i, b_i, i \in [r]$. Finally, for each round i, the verifier checks

$$\begin{aligned} &\textbf{if } b_i = 0, \quad c_0^{(i)} = Com(\mathbf{r}_i, \alpha\mathbf{r}_i - \mathbf{t}_1^{(i)}, \alpha\mathcal{F}(\mathbf{r}_i) - \mathbf{e}_1^{(i)}) \\ &\textbf{if } b_i = 1, \quad c_1^{(i)} = Com(\mathbf{r}_i, \alpha(\mathbf{v} - \mathcal{F}(\mathbf{r}_i) - \mathcal{G}(\mathbf{t}_1^{(i)}, \mathbf{r}_i) - \mathbf{e}_1^{(i)}) \end{aligned}$$

The signature is valid if all checks pass. To chose the right r for a security of λ bits, the number of rounds is computed as the smallest r such that

$$\frac{1}{\Pr[N]} + 2^{r-N} \geq 2^{\lambda}, \ \forall\ 0 \leq N \leq r$$

where

$$\Pr[N] = \sum_{i=N}^{r} \left(\frac{1}{q}\right)^i \left(\frac{q-1}{q}\right)^{r-i} \binom{r}{i}.$$

The main security property that is proven in [CHR+16] is the *existential unforgeability under adaptive chosen message attacks* (EU-CMA).

Theorem 3 (EU-CMA security)**.** *If $\mathcal{MQ}(n, m, \mathbb{F}_q)$ is intractable, if the hash functions $\mathsf{H}, \mathsf{H}_1, \mathsf{H}_2$ are modelled as random oracles and if Com is computationally binding and hiding, then* MQDSS *is* EU-CMA*-secure.*

Note that MQDSS made it to the second submission round of the NIST call for post-quantum digital signature scheme standardisation process.

Having exposed post-quantum hard problems, our focus now shifts to the exploration of various candidate constructions for anonymous tokens. These constructions derive their foundations from the aforementioned post-quantum challenges, paving the way for a comprehensive examination of possible anonymous token constructions.

3.2 Post-quantum anonymous credentials schemes

An anonymous credential scheme can easily be adapted to form an anonymous token scheme: the client requests a new credential as a token that it can only redeem once. Lately, [BLNS23] and [PWFHW23] came with some new constructions for a post-quantum anonymous credential scheme that we will briefly recall in the next sections. The former is based on the ISIS_f problem as the latter is based on the CRYSTALS Dilithium signature scheme. Both described some metrics, that we will also recall along the way, where **sig** denotes the signature size, **trans** denotes the transcript size of the issuing protocol, **prover** denotes the time for a user to prove the validity of a token (in ms) and finally **verifier** denotes the time for a verifier to check the validity of a token (in ms). We use a "-" when the metric is not available.

3.2.1 Post-quantum Privacy Pass

[PWFHW23] defines a post-quantum version of Privacy Pass. It is based on *zkDilithium*[6], a STARK[7]-friendly version of Dilithium2 which is currently in a standardisation process by NIST for becoming the standard post-quantum digital signature algorithm. The principle is the following:

- The client sends a commitment com of the attributes $(a_1, \ldots, a_k)$ to the issuer. Those attributes are used to include a nonce, timestamps, and the number of times the client used a token in the one day window[8].
- The issuer produces a signature σ on the commitment and sends it back to the user.
- To redeem the token, the user provides a zero-knowledge non-interactive argument of knowledge (zkNIAoK) of a signature σ on a commitment com and an opening of this commitment r to a set of k attributes $(a_1, ..., a_k)$.

The claim is that if all the algorithms used during this process are post-quantum resistant, then the scheme is post-quantum resistant. The protocol issues only one token at a time; this is because the token is multi-use per origin: each time a token is redeemed to an origin, the origin sends back a new valid token for the next use. They proposed three different implementations, one which is time-optimised, one which is size-optimised and a balanced one. We present their different metrics in Table 1.

scheme	tokens	λ	sig	trans	prover	verifier
[PWFHW23] better size	1	115	85.6 kB	2.4 kB	4882	19.8
[PWFHW23] balanced	1	115	112,3 kB	2.4 kB	660	22
[PWFHW23] better time	1	115	173.3 kB	2.4 kB	304	31

Table 1: *Metrics for the Post-quantum Privacy Pass protocol.*

We also recall that their implementation does not currently support computing proofs with zero-knowledge as the library they are using did not permit it at the time of the publication, but they stated that this should only add little overheads. Also the prover and verifier's time are mainly due to the zero-knowledge argument of knowledge used to prove knowledge of a token.

3.2.2 Practical anonymous credentials from lattices

[BLNS23] develops a post-quantum anonymous credential scheme that is based on lattices, and particularly the ISIS_f problem. We give below a very high-level overview of the scheme. First, the signer has to create its public and private key. To do so, it samples a random matrix A and a trapdoor which allows him to sample $\mathbf{s}$ such that $A\mathbf{s} = \mathbf{t}$ for any $\mathbf{t} \in \mathbb{Z}_p^n$. It finally creates two matrices B_1, B_2 and sets its public key to (A, B_1, B_2, f) and its private key to the trapdoor for matrix A. When the user wants a signature on a message $\mathbf{m}$, it samples a random vector $\mathbf{r}$, computes $\mathbf{c} := B_1\mathbf{m} + B_2\mathbf{r}$ and generates a proof that $\mathbf{m}$ and $\mathbf{r}$ are small and that $\mathbf{c}$ has been computed correctly. It sends $\mathbf{c}$ and the proof to the signer, who checks the proof, generates a tag $x \in [N]$ and uses the trapdoor to create $\mathbf{s}$ such that $A\mathbf{s} = f(x) + \mathbf{c}$. The signature is the

[6] https://github.com/guruvamsi-policharla/zkdilithium

[7] STARK stands for Scalable Transparent ARgument of Knowledge.

[8] Actually, this is three different attributes, with one in the 5 minutes window, one in the one hour window and one in the one day window, all respective to when the last time the token was (re)issued

tuple $(\mathbf{s}, x)$. To prove that it has a valid signature, the client reveals $\mathbf{m}$ and a zero-knowledge proof that the above equation is satisfied.

The schemes in [BLNS23] also issues one token at a time and can be used in the same way presented before for multi-use; recall that the current construction works for 8 credential attributes. They proposed two different constructions, one which is just based on the NTRU-ISIS$_f$ assumption and the other based on the Int-NTRU-ISIS$_f$ assumption, which is a reduction of the former and where it is assumed that it is as hard. We summarise the different metrics they obtained in Table 2.

scheme	tokens	λ	sig	trans	prover	verifier
[BLNS23] NTRU-ISIS$_f$	1	128	243 kB	473 kB	-	-
[BLNS23] Int-NTRU-ISIS$_f$	1	128	62 kB	107 kB	-	-

Table 2: *Metrics for the AnonCred protocol. No implementation has been made and thus we couldn't report the prover and verifier metrics.*

The first one does not look appealing due to the different sizes but the second one could be investigated further. As a matter of fact, size is not the only factor here but the timings of the proofs are also very important. Unfortunately, no implementation has been made by the authors and we thus have no point of comparison for the timings.

3.3 New post-quantum anonymous tokens constructions

A lot of research has been made in post-quantum VOPRFs. An anonymous token scheme can easily be built from such a primitive: one could sample a random tag in $\{0,1\}^\lambda$, hash it into the preimage set of the VOPRF construction, use the VOPRF to get a signature on the tag and then send the tag along with the signature to redeem the token. The verifiability is not directly given from the nature of the different schemes, but could be added with the help of zero-knowledge proofs. However we note that those proofs could add a lot of overhead for the sizes of the signatures and transcripts. We can also build post-quantum anonymous token schemes from other post-quantum primitives such as blind signatures and we also study this point further.

3.3.1 Post-quantum VOPRFs

[Bas23, dSGP23, ADDG23] all define post-quantum VOPRFs. The first two schemes are based on isogenies and the last one on homomorphic encryption on lattices. [Bas23] OPRF is inspired from the 2020 isogeny-based OPRF from Boneh, Kogan and Woo [BKW20] and is using M-SIDH [FMP23] as a building block. [dSGP23] builds an OPRF from group-action based cryptography, utilising new proof techniques they develop. Their construction can be instantiated with isogenies by using CSIDH [CLM+18]. We state their different metrics in Table 3 keeping the same labels as before but where the verifier's time replaced with the issuer's time and each metric is for the retrieval of one token.

The principal advantages of those post-quantum VOPRFs schemes is that they have a really small signature size. The verification would be only to send the token and the signature and verify the PRF on the server side which does not add a lot of overhead. Their main disadvantage is that they take a lot of bandwidth during the issuing phase. [ADDG23] would be a good candidate if it did not have to send a public key of 4077.8 MB with each issuing request. The last one [dSGP23] is also a good candidate and could be taken further to analyse the different running times.

scheme	λ	sig	trans	user	issuer
[Bas23]	128	32 B	8.7 MB	-	-
[ADDG23]*	128	32 B	4078,44 MB	256,1 ms	123ms**
[ADDG23]* 64-queries amortised	128	2048 B	4078,41 MB	256,1 ms	123ms**
[dSGP23]	λ	-	$(2\lambda + \frac{17}{2}) \log p + 4\lambda$	-	-

Table 3: *Comparison table for post-quantum VOPRFs primitives. No implementation has been made for [Bas23] and [dSGP23]; we thus do not have the metrics.*
* *Implementation has not been made with verifiability so timings might be wrong.*
** *Estimated.*

3.3.2 Post-quantum blind signatures

An anonymous token construction can easily be derived from a blind signature scheme: we first hash a tag $t \in \{0,1\}^\lambda$ into the message domain of the blind signature scheme; the issuing phase is the interactive signing phase and a token is the pair (t, σ) where σ is the blind signature; to verify the validity of a token the verifier hashes the tag t into the message domain and checks the blind signature on this message. [PSM17] proposed a blind signature scheme MBSS based on the hardness of the $\mathcal{MQ}$ problem, using the Rainbow and MQDSS building blocks. The protocol works as follows: let $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and $\mathcal{R} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ be two multivariate quadratic systems, with $\mathcal{P}$ being a Rainbow public key as described above and $\mathcal{R}$ being a random multivariate quadratic system. To get a blind signature for a message $\mathbf{w} \in \mathbb{F}^m$, the user samples a random value $\mathbf{z}^* \leftarrow\$ \mathbb{F}_q^m$, computes $\tilde{\mathbf{w}} := \mathbf{w} - \mathcal{R}(\mathbf{z}^*)$ and sends $\tilde{\mathbf{w}}$ to be signed. The issuer then returns a Rainbow signature $\mathbf{z}$ on $\tilde{\mathbf{w}}$ and the user ends up with a solution $(\mathbf{z}, \mathbf{z}^*)$ of the system $\mathcal{P}(\mathbf{x_1}) + \mathcal{R}(\mathbf{x_2}) = \mathbf{w}$. It can then prove the knowledge of this solution using MQDSS. Using the hash-and-sign paradigm described before, we can easily turn this blind signature protocol into an anonymous token scheme. The metrics of the blind signature scheme are summarised in Table 4. Their implementation was made with Sage and is thus not optimised. They claim that as their bottleneck is on the MQDSS, an optimised implementation as in [CHR+16] could drop the user and verifier timings by several orders of magnitude ([CHR+16] achieves MQDSS signatures of 256 bits security in 6.79 ms).

scheme	tokens	λ	sig	trans	user	issuer	verifier
[PSM17]	1	128	28.5 kB	-	7760	19	5505

Table 4: *MBSS metrics. The transcript size is equal to the sum of the byte-length of the encodings of vectors $\tilde{\mathbf{w}} \in \mathbb{F}_q^m$ and $\mathbf{z} \in \mathbb{F}_q^n$.*

As previously stated, the Rainbow post-quantum digital signature scheme has been unfortunately proven insecure, and thus an anonymous token construction can not be built on top of this blind signature scheme.

4 Multivariate Quadratic Anonymous Tokens

In this section, we present a new construction for post-quantum anonymous tokens based on the hardness of solving multivariate quadratic equation systems. The idea is pretty simple: we propose a modification of MBSS, which uses UOV instead of Rainbow as the first part of the signature, and use the transformation described in the previous section to get an anonymous token scheme. We have chosen to go further with this construction, as it could have significant results when optimised. After having presented this new construction, we prove that it conforms to the *unlinkability* and *one-more unforgeability* properties defined in Section 2.2. We then

discuss the choice of the parameters that can lead to post-quantum security.

4.1 The scheme

We call our new post-quantum anonymous token construction MQAT for **M**ultivariate **Q**uadratic **A**nonymous **T**okens.

Setup Algorithm. The $\mathsf{Setup}(\cdot)$ procedure is in charge of generating the scheme parameters depending on the security parameter λ. They are composed of:

- a finite field $\mathbb{F}_q$, where q is either a prime number or a prime power,
- integers $m, n \in \mathbb{N}$ such that the security level of the $\mathcal{MQ}$ instance $\mathcal{MQ}(m, n, \mathbb{F}_q)$ is greater or equal to λ,
- $r \in \mathbb{N}$, the numbers of rounds for MQDSS to have a security of λ bits (see Section 3.1.5),
- cryptographic hash functions
  - $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{2\lambda}$,
  - $\mathsf{H}_m : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \mathbb{F}_q^m$,
  - $\mathsf{H}_1 : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \mathbb{F}_q^r$ and
  - $\mathsf{H}_2 : \{0,1\}^\lambda \times \{0,1\}^\lambda \times (\mathbb{F}_q^{m+n})^r \to \{0,1\}^r$,
- two string commitment functions
  - $Com_0 : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^m \to \{0,1\}^{2\lambda}$ and
  - $Com_1 : \mathbb{F}_q^n \times \mathbb{F}_q^m \to \{0,1\}^{2\lambda}$.

Key Generation. The signer generates an UOV public/secret key pair, composed of an invertible linear transformation $\mathcal{T} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ and a set of m OV-polynomials $\mathcal{F} = (f^{(1)}, \ldots, f^{(m)})$ in n variables, and a random multivariate quadratic system $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$. The secret key consists of the pair $(\mathcal{F}, \mathcal{T})$ and the public key is composed of the two multivariate quadratic systems $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$ and $\mathcal{R}$.

Token issuance. The interactive token issuance protocol is depicted in Figure 8. The user first samples a random tag $t \in \{0,1\}^{2\lambda}$ that it uses to generate $\mathbf{w} \leftarrow \mathsf{H}_m(t) \in \mathbb{F}_q^m$. It then samples a random element $\mathbf{z}^* \leftarrow\!\!\$\ \mathbb{F}_q^m$, computes $\mathbf{w}^* := \mathcal{R}(\mathbf{z}^*)$ and sends $\tilde{\mathbf{w}} := \mathbf{w} - \mathbf{w}^*$ to the server. The signer sends back the UOV signature $\mathbf{z}$ of $\tilde{\mathbf{w}}$, which in fact is the preimage of $\tilde{\mathbf{w}}$ under the system $\mathcal{P}$. The user verifies the signature, checks that $\mathcal{P}(\mathbf{z}) + \mathcal{R}(\mathbf{z}^*) = \mathbf{w}$ and aborts if the check does not pass. It finally computes an MQDSS signature σ for the message $\mathbf{w}$ on the system $\bar{\mathcal{P}} = \mathcal{P} + \mathcal{R} : \mathbb{F}_q^{m+n} \to \mathbb{F}_q^m$, with $\mathcal{G}$ being the polar form of $\bar{\mathcal{P}}$. By doing this, the user proves that it knows a solution $(\mathbf{z}, \mathbf{z}^*)$ of the multivariate equations system $\mathcal{P}(\mathbf{x_1}) + \mathcal{R}(\mathbf{x_2}) = \mathbf{w}$ without revealing $\mathbf{z}$ and $\mathbf{z}^*$. Finally, the user stores the token $\mathsf{t} = (t, \sigma)$.

The token issuance protocol

User		**Issuer**
$t \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$		
$\mathbf{w} \leftarrow \mathsf{H}_m(t)$		
$\mathbf{z}^* \leftarrow\!\!\$\ \mathbb{F}_q^m$		
$\mathbf{w}^* := \mathcal{R}(\mathbf{z}^*)$		
$\tilde{\mathbf{w}} := \mathbf{w} - \mathbf{w}^*$		
	$\xrightarrow{\tilde{\mathbf{w}}}$	
		Find $\mathbf{u} \in \mathbb{F}_q^n$ such that $\mathcal{F}(\mathbf{u}) = \tilde{\mathbf{w}}$
		$\mathbf{z} := \mathcal{T}^{-1}(\mathbf{u})$
	$\xleftarrow{\mathbf{z}}$	
if $\mathcal{P}(\mathbf{z}) + \mathcal{R}(\mathbf{z}^*) \neq \mathbf{w}$ **then**		
abort		
$\mathcal{C} \leftarrow \mathsf{H}(\mathcal{P}\|\|\mathbf{w})$		
$\mathcal{D} \leftarrow \mathsf{H}(\mathcal{C}\|\|\mathbf{w})$		
$\mathbf{r}_0^{(0)}, \ldots, \mathbf{r}_0^{(r-1)} \leftarrow\!\!\$\ \mathbb{F}_q^{m+n}$		
$\mathbf{t}_0^{(0)}, \ldots, \mathbf{t}_0^{(r-1)} \leftarrow\!\!\$\ \mathbb{F}_q^{m+n}$		
$\mathbf{e}_0^{(0)}, \ldots, \mathbf{e}_0^{(r-1)} \leftarrow\!\!\$\ \mathbb{F}_q^{m}$		
$\mathbf{r}_1^{(i)} := (\mathbf{z}\|\|\mathbf{z}^*) - \mathbf{r}_0^{(i)}$		
$c_0^{(i)} := Com_0(\mathbf{r}_0^{(i)}, \mathbf{t}_0^{(i)}, \mathbf{e}_0^{(i)})$		
$c_1^{(i)} := Com_1(\mathbf{r}_1^{(i)}, \mathcal{G}(\mathbf{t}_0^{(i)}, \mathbf{r}_1^{(i)}) + \mathbf{e}_0^{(i)})$		
$\sigma_0 := \mathsf{H}(c_0^{(0)}\|\|c_1^{(0)}\|\| \ldots \|\|c_0^{(r-1)}\|\|c_1^{(r-1)})$		
$\alpha_0, \ldots, \alpha_{r-1} \leftarrow \mathsf{H}_1(\mathcal{D}, \sigma_0)$		
$\mathbf{t}_1^{(i)} := \alpha_i \cdot \mathbf{r}_0^{(i)} - \mathbf{t}_0^{(i)}$		
$\mathbf{e}_1^{(i)} := \alpha_i \cdot \bar{\mathcal{P}}(\mathbf{r}_0^{(i)}) - \mathbf{e}_0^{(i)}$		
$\sigma_1 := (\mathbf{t}_1^{(0)}, \mathbf{e}_1^{(0)}, \ldots, \mathbf{t}_1^{(r-1)}, \mathbf{e}_1^{(r-1)})$		
$b_0, \ldots, b_{r-1} \leftarrow \mathsf{H}_2(\mathcal{D}, \sigma_0, \sigma_1)$		
$\sigma_2 := (\mathbf{r}_{b_0}^{(0)}, \ldots, \mathbf{r}_{b_{r-1}}^{(r-1)}, c_{1-b_0}^{(0)}, \ldots, c_{1-b_{r-1}}^{(r-1)})$		
$\sigma := (\mathcal{C}, \sigma_0, \sigma_1, \sigma_2)$		
return (t, σ)		

Figure 8: *The issuance protocol for* MQAT*, our new anonymous tokens scheme.*

Verification algorithm. The token verification protocol is depicted in Figure 9. On receiving the token $\mathsf{t} = (t, \sigma)$, the verifier computes $\mathbf{w} \leftarrow \mathsf{H}_m(t)$, checks that the MQDSS signature σ on $\mathbf{w}$ is valid and outputs the result as a boolean value.

$\mathsf{MQAT.Verify}(\mathsf{pk} = (\mathcal{P}, \mathcal{R}), \mathsf{t} = (t, \sigma))$

$(\mathcal{C}, \sigma_0, \sigma_1, \sigma_2) := \sigma$
$\bar{\mathcal{P}} = \mathcal{P} + \mathcal{R}$
$\mathbf{w} \leftarrow \mathsf{H}_m(t)$
$\mathcal{D} \leftarrow \mathsf{H}(\mathcal{C}||\mathbf{w})$
$\alpha_0, \ldots, \alpha_{r-1} \leftarrow \mathsf{H}_1(\mathcal{D}, \sigma_0)$
$b_0, \ldots, b_{r-1} \leftarrow \mathsf{H}_2(\mathcal{D}, \sigma_0, \sigma_1)$
for $i = 0, \ldots, r-1$ **do**
  Extract $(\mathbf{t}_1^{(i)}, \mathbf{e}_1^{(i)})$ from σ_1 and $\mathbf{r}_i$ from σ_2
  if $b_i = 0$ **then**
    $c_0^{(i)} := Com(\mathbf{r}_i, \alpha\mathbf{r}_i - \mathbf{t}_1^{(i)}, \alpha\bar{\mathcal{P}}(\mathbf{r}_i) - \mathbf{e}_1^{(i)})$
  if $b_i = 1$ **then**
    $c_1^{(i)} := Com(\mathbf{r}_i, \alpha(\mathbf{w} - \bar{\mathcal{P}}(\mathbf{r}_i)) - \mathcal{G}(\mathbf{t}_1^{(i)}, \mathbf{r}_i) - \mathbf{e}_1^{(i)})$
$\sigma_0' := \mathsf{H}(c_0^{(0)}||c_1^{(0)}||\ldots||c_0^{(r-1)}||c_1^{(r-1)})$
return $\sigma_0 = \sigma_0'$

Figure 9: *The verification algorithm of* MQAT.

Notes. To reduce the length of the signature, we use techniques mentioned in [CHR+16] and [PSM17]. Instead of including the concatenation of the $2r$ commitments directly into the signature, we set $\sigma_0 := \mathsf{H}(c_0^{(0)}||c_1^{(0)}||\ldots||c_0^{(r-1)}||c_1^{(r-1)})$. This comes with the overhead of having to transmit the commitments $(c_{1-b_0}^{(0)}, \ldots, c_{1-b_{r-1}}^{(r-1)})$ in σ_2, and the verifier having to recompute the commitments $(c_{b_0}^{(0)}, \ldots, c_{b_{r-1}}^{(r-1)})$. We thus saved having to send r commitments.
Another note is that the private key of the issuer is never used in the verification procedure. This then makes the protocol publicly verifiable.

The proof for correctness follows the one from [PSM17]. We split the proof in two parts. The first step shows that the user rightfully ends up with a solution $(\mathbf{z}, \mathbf{z}^*)$ of the system $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{w}$. This is true since by definition $\mathcal{P}(\mathbf{z}) = \tilde{\mathbf{w}}$, $\mathcal{R}(\mathbf{z}^*) = \mathbf{w}^*$ and $\mathbf{w} = \tilde{\mathbf{w}} + \mathbf{w}^*$. The second step shows by the correctness of the MQDSS scheme defined in [CHR+16] that the user knows a the solution of the system $\bar{\mathcal{P}}(\mathbf{x}) = \mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2)$.

4.2 Security

In this section, we discuss and prove the *unlinkability* and *one-more unforgeability* properties of our new scheme MQAT. We then analyse the notion of *post-quantum security* for the primitives used in the construction.

Unlinkability. In the UNLINK game, the adversary needs to find which query $\tilde{\mathbf{w}}_i$ is linked to the output token $(t, \sigma)_j$ selected randomly. The signature σ does not contain any information about the solution $(\mathbf{z}, \mathbf{z}^*)$ of the system $\mathcal{P}(\mathbf{x_1}) + \mathcal{R}(\mathbf{x_2}) = \mathbf{w}$ due to the zero-knowledge property of MQDSS. Therefore, the adversary has no choice but to link a query $\tilde{\mathbf{w}}_\mathbf{i}$ to t as σ does not give any advantage. We know that $\tilde{\mathbf{w}} = \mathbf{w} - \mathcal{R}(\mathbf{z}^*)$ for a vector $\mathbf{z}^* \leftarrow\!\!\$\ \mathbb{F}_q^m$ and $\mathbf{w} = \mathsf{H}_m(T)$. As a result, the problem reduces to finding this $\mathbf{z}^*$ for the queries $\tilde{\mathbf{w}}_i$ made by the adversary in $\mathcal{A}_1$. As the $\mathcal{MQ}$ problem is intractable and as $\mathbf{z}^*$ is uniformly chosen in $\mathbb{F}_q^m$, then the best solution for the adversary is no better than guessing on the $\tilde{\mathbf{w}}_i$'s, which corresponds to the 1-*unlinkability* property.

Theorem 4 (MQAT is 1-unlinkable)**.** MQAT *is* 1-unlinkable *if*

- $\mathcal{MQ}(n, m, \mathbb{F}_q)$ *is intractable,*
- Com_0 *and* Com_1 *are statistically hiding commitment schemes,*
- $\mathsf{H}, \mathsf{H}_m, \mathsf{H}_1$ *and* H_2 *are modelled as random oracles,*
- *and the distribution of* $\mathcal{R}(\mathbf{x})$ *for* $\mathbf{x} \leftarrow\!\!\$\ \mathbb{F}_q^m$ *is computationally indistinguishable from uniform in* $\mathbb{F}_q^m$*.*

Proof. Let **Game 0** be the UNLINK game as in Definition 2. Let **Game 1** be the UNLINK game as **Game 0** but where the functions $\mathsf{AT.User}_0(\cdot)$ and $\mathsf{AT.User}_1(\cdot)$ are replaced by their instantiated algorithms $\mathsf{MQAT.User}_0(\cdot)$ and $\mathsf{MQAT.User}_1(\cdot)$ as defined in Figure 8. As this is the same game, the advantage of $\mathcal{A}$ does not change. So we have that

$$\mathsf{Adv}^{\mathsf{UNLINK}}_{\mathcal{A},\ell}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A},\ell}(\lambda).$$

Let **Game 2** be the same as **Game 1** but where $\mathbf{w}^*$ in $\mathsf{MQAT.User}_0()$ is sampled uniformly at random in $\mathbb{F}_q^m$ rather by being computed with $\mathcal{R}(\cdot)$, as by assumption the output of $\mathcal{R}(\mathbf{x})$ is uniformly distributed in $\mathbb{F}_q^m$ for any $\mathbf{x} \in \mathbb{F}_q^m$. We can then also remove the sampling of $\mathbf{z}^*$. Thus the advantage of the adversary after $\mathcal{A}_1$ is bounded by the capability of a distinguisher $\mathcal{D}$ to distinguish between a random value and the output of $\mathcal{R}(\cdot)$, which is by assumption negligible. Hence

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A},\ell}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A},\ell}(\lambda) + \Pr[\mathcal{D}\ wins].$$

Let **Game 3** be defined as **Game 2** but instead of sampling $\mathbf{w}^*$ we directly sample $\tilde{\mathbf{w}}$ as it has the same distribution as $\mathbf{w}^*$ since $\mathbf{w}^*$ is sampled uniformly at random and perfectly hides $\mathbf{w}$. At this point, the adversary has no way to link t and $\tilde{\mathbf{w}}$, as they are completely independent. Thus,

$$\mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A},\ell}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A},\ell}(\lambda).$$

Let **Game 4** be as **Game 3** but where the procedure $\mathsf{MQAT.User}_1(\cdot)$ is replaced by the MQDSS zero-knowledge simulator $\mathcal{M}$. As by assumption the commitment schemes are *statistically hiding*, the underlying 5-pass identification scheme is *statistically zero-knowledge* ([SSH11], Section 4, Theorem 4) and thus so is MQDSS. As a consequence, the capability of an adversary $\mathcal{B}$ to distinguish between an honest MQDSS signature and a simulated one, which is also negligible, is added to the advantage. Hence,

$$\mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A},\ell}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A},\ell}(\lambda) + \Pr[\mathcal{B}\ wins].$$

At this point, the view of the adversary in **Game 4** is composed of ℓ random $\tilde{\mathbf{w}}_i$ and ℓ random (t_j, σ_j) all independent of the $\tilde{\mathbf{w}}_i$'s, and hence the best strategy for an adversary of **Game 4** is to chose an index $j' \leftarrow\!\!\$\ [\ell]$ uniformly at random. So

$$\mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A},\ell}(\lambda) = \frac{1}{\ell}.$$

Putting everything together, we have that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{UNLINK}}_{\mathcal{A},\ell}(\lambda) &\leq \frac{1}{\ell} + \Pr[\mathcal{D}\ wins] + \Pr[\mathcal{B}\ wins] \\ &= \frac{1}{\ell} + \mathsf{negl}(\lambda)\end{aligned}$$

as by assumptions the last two terms on the right-hand side are negligible. We can then conclude that MQAT is 1-*unlinkable*. □

Unforgeability. In the OMUF game, the adversary needs to provide $\ell + 1$ valid signatures, having access to ℓ queries to a signing oracle. This security notion relies heavily on the unforgeability properties of both UOV and MQDSS, as being able to forge a signature of either scheme, an adversary could easily produce valid tokens.

Theorem 5 (MQAT is one-more unforgeable)**.** MQAT *is* one-more token unforgeable *if*

- $\mathcal{MQ}(n, m, \mathbb{F}_q)$ *is intractable,*
- Com_0 *and* Com_1 *are computationally binding commitment schemes,*
- MQDSS *is* EU-CMA-secure*,*
- *finding a solution* $(\mathbf{z}, \mathbf{z}^*)$ *of the multivariate quadratic equation system* $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{0}$ *for a randomly chosen quadratic map* $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ *and an* UOV *public key* $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ *is a hard problem,*
- UOV *is secure,*
- *and the distribution of* $\mathcal{R}(\mathbf{x})$ *for* $\mathbf{x} \leftarrow\!\!\$\ \mathbb{F}_q^m$ *is computationally indistinguishable from uniform in* $\mathbb{F}_q^m$*.*

Proof. Consider an adversary $\mathcal{A}$ against OMUF of MQAT. We present by a sequence of game-hopping arguments that an adversary winning the OMUF game logically implies that there exist an adversary that can find a solution $(\mathbf{z}, \mathbf{z}^*)$ of the multivariate quadratic equation system $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{0}$ for a randomly chosen quadratic map $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and an UOV public key $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, which is a hard problem. Along the proof, we will denote by $\mathcal{A}_i$ an adversary against **Game i**.

Let **Game 0** be the OMUF game as in Definition 3, instantiated with the algorithms of MQAT defined in Section 4.1. Let **Game 1** be the OMUF game, but with the $\mathsf{Verify}(\cdot)$ procedure being replaced by its interactive counterpart, as defined in [SSH11], and the random oracles that $\mathcal{A}_0$ uses are programmed to respond the challenges that $\mathcal{A}_1$ receives from the challenger. Clearly, $\mathcal{A}_0$ winning implies $\mathcal{A}_1$ winning and thus

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathcal{A},\ell}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}_1,\ell}(\lambda).$$

Let **Game 2** be the OMUF game where the zero-knowledge property in the verification procedure is dropped; that is the adversary directly sends $(\mathbf{z}, \mathbf{z}^*)$ to the challenger. $\mathcal{A}_2$ uses the knowledge extractor $\mathcal{E}$ of the 5-pass zero-knowledge identification scheme to recover $(\mathbf{z}, \mathbf{z}^*)$ from the outputs of $\mathcal{A}_1$ and wins the game. Recall from Theorem 2 that the extractor $\mathcal{E}$ outputs a solution with probability $\frac{q+1}{2q}$, and that MQDSS uses a parallel composition of r rounds of the 5-pass identification scheme from [SSH11]. So we have that

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}_1,\ell}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}_2,\ell}(\lambda) + \left(\frac{q+1}{2q}\right)^r.$$

Let **Game 3** be defined as the following game:

Game 3(λ) :

1 : $\mathsf{MQAT.Setup}(1^\lambda) \to \mathsf{cpp}$
2 : $\mathsf{MQAT.KeyGen}(\mathsf{cpp}) \to (\mathsf{sk}, \mathsf{pk})$
3 : $queries := \emptyset$
4 : $inverses := \emptyset$
5 : $(t, (\mathbf{z}, \mathbf{z}^*)) \leftarrow \mathcal{A}_3^{\mathsf{OIssue},\mathsf{OInv}}(\mathsf{cpp}, \mathsf{pk})$
6 : **if** $t \in queries$ **or**
7 : $\quad \mathbf{z} \in inverses$ **then abort**
8 : **return** $\mathcal{P}(\mathbf{z}) + \mathcal{R}(\mathbf{z}^*) = \mathsf{H}_m(t)$

OIssue()

1 : $(t, \sigma) \leftarrow \langle \mathsf{MQAT.User}(\mathsf{pk}), \mathsf{AT.Sign}(\mathsf{sk}) \rangle$
2 : $queries := queries \cup \{t\}$
3 : **return** (t, σ)

OInv(query)

1 : $\mathsf{resp} \leftarrow \mathsf{MQAT.Sign}_0(\mathsf{sk}, \mathsf{query})$
2 : $inverses := inverses \cup \{\mathsf{resp}\}$
3 : **return** resp

Adversary $\mathcal{A}_3$ calls $\mathcal{A}_2$ with its inputs and when $\mathcal{A}_2$ calls oracle OSign $\mathcal{A}_3$ calls oracle OInv. When $\mathcal{A}_2$ outputs its list of $\ell+1$ tokens, $\mathcal{A}_3$ outputs the token for which $\mathbf{z}$ was not already seen and wins the game. Hence,

$$\mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}_2,\ell}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}_3}(\lambda).$$

Let **Game 4** be the same game as **Game 3**, but without the inverses oracle OInv. It is clear that an adversary winning **Game 3** will win **Game 4**: as UOV is secure by assumption, seeing inverses does not give any advantage to adversary $\mathcal{A}_4$. Thus,

$$\mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}_3}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A}_4}(\lambda).$$

Let **Game 5** be the following game: given $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ an UOV public key, $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ a randomly chosen quadratic map and access to an oracle $\mathsf{H}'(x) = \mathcal{P}(\mathsf{H}_n(x)) + \mathcal{R}(\mathsf{H}_m(x))$, where $\mathsf{H}_n : \{0,1\}^* \to \mathbb{F}_q^n$ and $\mathsf{H}_m : \{0,1\}^* \to \mathbb{F}_q^m$ are true random oracles, find a tuple $(t, (\mathbf{x}_1, \mathbf{x}_2))$ such that $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathsf{H}_m(t)$. An adversary winning **Game 4** wins **Game 5**, so

$$\mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A}_4}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}_5,\mathcal{P},\mathcal{R}}(\lambda).$$

Let **Game 6** be the following game: given $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ an UOV public key, $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ a randomly chosen quadratic map and $\mathbf{w} \leftarrow\!\$\, \mathbb{F}_q^m$, find a tuple $(\mathbf{x}_1, \mathbf{x}_2)$ such that $\mathcal{P}(\mathbf{x}_1)+\mathcal{R}(\mathbf{x}_2) = \mathbf{w}$. Since the output of $\mathcal{R}(\mathbf{x})$ is indistinguishable from a random value in $\mathbb{F}_q^m$, the output of the oracle $\mathsf{H}'(x)$ does not give any advantage, and since H_m is modelled as a random oracle, an adversary winning **Game 5** will win **Game 6**. Hence,

$$\mathsf{Adv}^{\mathsf{Game}_5}_{\mathcal{A}_5,\mathcal{P},\mathcal{R}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_6}_{\mathcal{A}_6,\mathcal{P},\mathcal{R},\mathbf{w}}(\lambda).$$

Let **Game 7** be the following game: given $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ an UOV public key, $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ a randomly chosen quadratic map , find a tuple $(\mathbf{x}_1, \mathbf{x}_2)$ such that $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{0}$. $\mathcal{A}_7$ wins by simulating **Game 6** with $(\mathcal{P}, \mathcal{R}+\mathbf{w}, \mathbf{w})$ for a randomly chosen $\mathbf{w}$. Hence, putting everything together, we have that

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathcal{A},\ell}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_7}_{\mathcal{A}_7,\mathcal{P},\mathcal{R}}(\lambda) + \left(\frac{q+1}{2q}\right)^r.$$

As **Game 7** is exactly the same problem as *finding a solution* $(\mathbf{z}, \mathbf{z}^*)$ *of the multivariate quadratic equation system* $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{0}$ *for a randomly chosen quadratic map* $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ *and an* UOV *public key* $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, which by assumption no adversary can win with probability better than negligible, and r has been chosen such that MQDSS is $\mathsf{EU\text{-}CMA}$-*secure* in order for $\left(\frac{q+1}{2q}\right)^r$ to be negligible, thus finally

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathcal{A},\ell}(\lambda) \leq \mathsf{negl}(\lambda)$$

which concludes the proof that MQAT is *1-more unforgeable*. □

As it was also the case in [PSM17], one of the assumption of Theorem 5 remains to be shown: *finding a solution* $(\mathbf{z}, \mathbf{z}^*)$ *of the multivariate quadratic equation system* $\bar{\mathcal{P}}(\mathbf{x}) = \mathcal{P}(\mathbf{x}_1)+\mathcal{R}(\mathbf{x}_2) = \mathbf{0}$ *for a randomly chosen quadratic map* $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ *and an* UOV *public key* $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ *is a hard problem*. We do not provide a proof for the statement above, but rather justify this premise by relying on the common hardness of $\mathcal{MQ}$-based cryptography. There are two kinds of strategies to attack the above assumption: 1) directly trying to solve the system $\bar{\mathcal{P}}(\mathbf{x}) = \mathbf{0}$ as an instance of the $\mathcal{MQ}$ problem and 2) use the special structure of system $\bar{\mathcal{P}}$ to decompose it into easily invertible maps. We will discuss below the classical and post-quantum algorithms for solving the $\mathcal{MQ}$ problem and then the structural attacks on $\bar{\mathcal{P}}$.

4.2.1 Algorithms for solving the $\mathcal{MQ}$ problem

The number of equations m and the number of variables n strongly determine the hardness of solving the $\mathcal{MQ}$ problem. Whenever $m \geq n$ (over-determined systems), we give away more information about the system and the attacks described below perform even better; there is then no incentive of choosing $m \geq n$. On the other hand, when $n > m$ (under-determined systems), one can simply fix the $n - m$ excessive variables and the problem thus reduces to solving a system of m equations in m variables; so there is also no incentive of choosing $n > m$. With the above remarks in mind, we will assume for the rest of this section that $m = n$. Also, without loss of generality, let $\mathcal{F}(\mathbf{x}) \in \mathcal{MQ}(n, m, \mathbb{F}_q)$ be the $\mathcal{MQ}$ problem instance that we are trying to solve.

Exhaustive search. The most naive and simple way of solving the $\mathcal{MQ}$ problem would be to try all the possible $\mathbf{x} \in \mathbb{F}_q^n$. For a single polynomial, this would result in $\mathcal{O}(n^2)$ additions and multiplications. The overall complexity would then be of $\mathcal{O}(mn^2 \cdot q^n)$ field operations. Another technique introduced in [BCC+10] for fast enumeration in $\mathbb{F}_2$ would only require $\log(n)2^{n+2}$ operations and could be generalised for larger field size, resulting in $\mathcal{O}(\log_q(n) \cdot q^n)$ operations for a field $\mathbb{F}_q$.

The hybrid approach. Currently, the best known techniques to solve multivariate quadratic equation systems is to use a combination of exhaustive search and quadratic equation solvers, such as the F5 algorithm [Fau02] or the FXL algorithm [YC05]. These techniques are called the hybrid approach and have been first presented in [BFP09]. The general idea of these algorithms is to fix k variables amongst the n, then use advanced computer algebra techniques, often called specialisation processes, to solve the smaller system in $n-k$ variables, and then finally optimise for this very parameter k.

Crossbred. The Crossbred algorithm has been developed in [JV18]. It was originally designed to work in binary fields ($\mathbb{F}_2$) but could be generalised in arbitrary sized fields $\mathbb{F}_q$. The principle is to perform first some operations on the system and only then fix the variables as above.

Quantum attacks. There is no specialised quantum algorithm yet designed to solve multivariate quadratic equation systems. One of the best promising way to speed up the attacks described above would be to use Grover's algorithm [Gro96]. In fact, it would outperform the above approaches by at most a square root factor. In [SW16], the authors described two different quantum circuits that could be used as oracles in Grover's algorithm.

Complexities of those algorithms are discussed in the next section, with some concrete numbers.

4.2.2 Structural attacks on $\bar{\mathcal{P}}$

The special structure of the quadratic multivariate equations system $\bar{\mathcal{P}}$ could be used to facilitate attacks against the aforementioned assumption. The goal of such attacks on $\bar{\mathcal{P}}$ is to find a decomposition in easily invertible maps. In the case of our system, we have that

$$\begin{aligned}\bar{\mathcal{P}}(\mathbf{x}) &= \mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) \\ &= \mathcal{F} \circ \mathcal{T}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) \\ &= (\mathcal{F} + \mathcal{R}) \circ \mathcal{T}'(\mathbf{x}) \\ &= \mathcal{F}' \circ \mathcal{T}'(\mathbf{x})\end{aligned}$$

where the linear transformation $\mathcal{T}'$ is represented by the matrix

$$T' = \begin{pmatrix} T & 0 \\ 0 & \mathbf{1}_m \end{pmatrix}.$$

Then, we need to recover this map $\mathcal{T}'$ using the known structure of $\mathcal{F}' = \mathcal{F} + \mathcal{R}$. As the coefficients of $\mathcal{R}$ have been chosen uniformly at random in $\mathbb{F}_q$ so that $\mathcal{R}$ is a random multivariate quadratic equations system, the only structure we can use to recover the matrix T is the central map $\mathcal{F}$. Thus, a structural attack against the structure of our multivariate quadratic system $\bar{\mathcal{P}}$ is at least as hard as a structural attack on system $\mathcal{P}$.

By choosing appropriate parameters for our new anonymous tokens construction, such that the combination of the underlying primitives (mainly UOV and MQDSS) is secure, we can prevent all the attacks described above.

5 Implementation

In this section we will present an concrete Go implementation of MQAT. The source code can be found at https://github.com/sebhauri/mqat. We first discuss the parameter selection according to the different attacks mentioned above and then explain different implementation details that are relevant. Finally, we state the performance results that we obtained.

5.1 Choice of the $\mathcal{MQ}$ parameters

Following the security analysis, we want to choose the best fitted parameters for the aimed security level. We first choose the parameters for UOV, as it is the first primitive that appears in our issuing scheme. As we want to achive NIST's first level of security, this leaves us with two choices of parameters[9]:

	q	**m**	**n**
uov-Ip	256	44	112
uov-Is	16	64	160

Table 5: *Parameters for NIST security level 1 of the* UOV *signature scheme taken from the official recommended parameter sets.*

The first one reduces slightly the size of the keys and the second one reduces slightly the signature size. Regarding their performance, the first one can sign and verify signatures in less cycles than the second one. Also, implementation of $\mathbb{F}_{256}$ seems easier to implement in typed programming languages, as one field element can be represented by one byte. Thus we decide to go with the first parameter set. We now need to see if those parameters can fit with the second part of our scheme, MQDSS. Recall that we add the system $\mathcal{R}$ of m equations in m variables on the UOV system, and so MQDSS is performed on the system $\bar{\mathcal{P}} = \mathcal{P} + \mathcal{R} : \mathbb{F}_q^{n+m} \to \mathbb{F}_q^m$, where (q, m, n) are the UOV parameters. So the parameters for MQDSS in our scheme are $q = 256$, $m = 44$ and $n = 158$. We then need to show that the best known attacks on the $\mathcal{MQ}$ problem are mitigated for this parameter set. In our case, $n > m$, which means that an adversary against the $\mathcal{MQ}$ problem could fix $n - m$ variables of the system and solve the $\mathcal{MQ}$ system for $q = 256$ and $m = n = 44$, as explained in the attacks against the $\mathcal{MQ}$ problem in Section 4.2.1. Fortunately, the MQDSS latest specification[10] (Specification 2.1, Section 2.2, Table 2.2) made the analysis for $q = 256$ and $m = n = 40$. The results are summarized in the table below:

[9] An official publication [BCH+23] has been made for UOV modern parameters. The most recent specifications can be found on https://www.uovsig.org/.

[10] The official publication [CHR+16] of MQDSS only specifies one parameter set. Most recent parameters can be found on https://mqdss.org/, under the latest specification.

	k	Field operations
FastEnum	-	2^{328}
HybridF5	3	2^{153}
Crossbred	23	2^{206}

Table 6: *Classical attack complexities on the $\mathcal{MQ}$ problem for parameters $n = m = 40, q = 256$. The value k represents the number of fixed variables in the different specialisation processes.*

As in our case $m = 44$, those parameters seem to be enough to get a security of $\lambda = 128$ bits. For quantum security, the latest specification (Specification 2.1, Section 2.3, Tables 2.3, 2.4 and 2.5) also gives an estimates of the different attacks using Grover's algorithm to speed up computations, and the results are presented it the table below:

			Gates		Circuit depth	
	n	**k**	**T**	**Clifford**	**T**	**Total**
FastEnum	32	-	$2^{152.21}$	$2^{152.69}$	$2^{142.63}$	$2^{143.83}$
HybridF5	40	19	$2^{107.80}$	$2^{145.27}$	$2^{97.44}$	$2^{98.51}$
	48	23	$2^{124.58}$	$2^{164.80}$	$2^{113.70}$	$2^{114.77}$
Crossbred	48	23	$2^{124.58}$	$2^{125.06}$	$2^{114.96}$	$2^{116.14}$

Table 7: *Quantum attack complexities on the $\mathcal{MQ}$ problem for parameters $q = 256$ with different values of n. The value k represents the number of fixed variables in the different specialisation processes. The count of gates and depths is given as the number of T-gates along with the overall number of Clifford gates.*

With the complexities described above, we are confident with our parameter set, namely $m = 44, n = 112$ and $q = 256$, that our new anonymous token scheme MQAT is secure against classical and post-quantum direct attacks against the underlying $\mathcal{MQ}$ problem.

5.2 Other parameters

We need all the other cryptographic primitives to be chosen in order to achieve a security of $\lambda = 128$ bits. The primitives left are composed of the different cryptographic hash functions, the commitment schemes and the pseudo-number generators. We also need to chose the number of rounds r in order for MQDSS to achieve EU-CMA-security.

The random oracles. For the hash function H and the two commitment schemes Com_0 and Com_1, we use the standard Go implementation of SHA3-256. From now on, we use the value `HASH_BYTES` to denote the output length of this function, in bytes. This ensures us a quantum security of 128 bits. For the hash functions $\mathsf{H}_m, \mathsf{H}_1$ and H_2, we use the standard Go implementation of SHAKE-256. This function allows us to create variable-output-length hash and achieves a generic security strength of 256 bits in the random oracle model and thus at least 128 bits of quantum security.

The number of MQDSS rounds. Another parameter to set now is the number r of rounds for the MQDSS to be sound enough. Recall that there is a formula for this parameter, that is the smallest r such that

$$\frac{1}{\Pr[N]} + 2^{r-N} \geq 2^{\lambda}, \ \forall\ 0 \leq N \leq r$$

where

$$\Pr[N] = \sum_{i=N}^{r} \left(\frac{1}{q}\right)^{i} \left(\frac{q-1}{q}\right)^{r-i} \binom{r}{i}$$

and λ is the security parameter. We aim at a security level $\lambda = 128$ to achieve NIST security level 1, and with our value $q = 256$ we find the number of round for MQDSS to be $r = 156$. The Python script to find this parameter is presented in Appendix A.

5.3 Implementation details

We discuss in this section other implementation details. First, we need a way to represent a field element and define its basic operations. We chose our field $\mathbb{F}_{256}$ to be the AES field $\mathbb{F}_2[X]/(X^8 + X^4 + X^3 + X + 1)$. Elements of $\mathbb{F}_{256}$ are then polynomials of degree less than 8 with coefficients in $\{0, 1\}$. We can easily represent them as bytes, where bit b_i represents the coefficient of degree i, $i = 0, \ldots, 7$. In our Go implementation, we use the unsigned integer over 8 bits basic type `uint8`. Having this defined, we need a way to compute quadratic multivariate polynomials. In our implementation, we only consider quadratic homogeneous polynomials, as this does not seem to lower the security of the underlying schemes. Then a multivariate quadratic polynomial $f_i(\mathbf{x})$ in n variables can be uniquely represented as an upper-diagonal matrix $P_i \in \mathbb{F}_{256}^{n \times n}$, such that

$$f_i(\mathbf{x}) = \mathbf{x}^T \cdot P_i \cdot \mathbf{x}$$

composed of $\frac{n \cdot (n+1)}{2}$ non-zero elements. A vector in $\mathbb{F}_{256}^{n}$ is represented as an n-byte string. In the next paragraphs, we will discuss implementation details about the different algorithms of our scheme.

Key generation. We recall that the private key of our anonymous token scheme is composed of an UOV private key and that the public key is the corresponding UOV public key along with a random multivariate quadratic system $\mathcal{R}$. We use the latest UOV specification ([BCH+23]) to implement the UOV part, following the `classic` variant, using some twists to simplify the implementation that we explain hereafter. The expansion of matrices $\{P_i^{(1)}\}_{i \in [m]}$ and $\{P_i^{(2)}\}_{i \in [m]}$ is implemented using the standard SHAKE-128 implementation as opposed to aes128ctr. We sample matrices in row-major order where for the particular case of upper-diagonal matrices we do not encode the zeroes below the diagonal. Set of matrices are encoded with the concatenation of matrices and not in the interleaved fashion. We use a seed of λ bits to generate the random system $\mathcal{R}$ and we recompute the system with the standard SHAKE-128 implementation whenever needed; we thus reduce the size of the public key. This random system is composed of m upper-triangle matrices in $\mathbb{F}_{256}^{m \times m}$. Overall, we have that

$$\mathsf{sk} = \left(\mathsf{seed\_uov\_sk}, O, \left\{P_i^{(1)}, S_i\right\}_{i \in [m]}\right),$$
$$\mathsf{pk} = \left(\mathsf{seed\_uov\_pk}, \left\{P_i^{(1)}, P_i^{(2)}, P_i^{(3)}\right\}_{i \in [m]}, \mathsf{seed\_random\_sys}\right)$$

with

$$|\mathsf{sk}| = \frac{\lambda}{8} + (m \cdot (n-m)) + \left(m \cdot \frac{(n-m)(n-m+1)}{2} + m^2 \cdot (n-m)\right) \text{ bytes},$$
$$|\mathsf{pk}| = \frac{\lambda}{8} + \left(m \cdot \frac{n \cdot (n+1)}{2}\right) + \frac{\lambda}{8} \text{ bytes}.$$

Token issuance. In the $\mathsf{User}_0(\cdot)$ procedure, $\mathbf{z}^*$ is generated using the Go standard SHAKE-256 implementation with a fresh randomly generated seed of 2λ bits. The output state is composed of the tag t and the computed value $\mathbf{z}^*$. The $\mathsf{Sign}_0(\cdot)$ procedure is implemented as in the latest UOV specification ([BCH+23]). The only difference is that we do not use the hash as the message is already in the correct domain. Thus we also do not use a salt to generate vector $\mathbf{v}$ (the random vinegar vector). Finally, the $\mathsf{User}_1(\cdot)$ expands the random system $\mathcal{R}$ from `seed_random_sys` as explained above and computes an MQDSS signature σ for vector $(\mathbf{z}||\mathbf{z}^*)$ of the system $\bar{\mathcal{P}} = \mathcal{P}+\mathcal{R}$. We follow the latest MQDSS specification (2.1) transposed for $\mathbb{F}_{256}$ to do so. The only differences is that we do not need to expand neither the $\mathcal{MQ}$ system nor the secret vector has it is already done. Overall, we end with a token $\mathsf{t} = (t, \sigma)$ with

$$|\mathsf{t}| = \frac{2\lambda}{8} + ((2+r) \cdot \mathsf{HASH\_BYTES} + r \cdot (2n+m)) \text{ bytes.}$$

Token verification. The token verification is implemented by first hashing the tag t into $\mathbf{w}$ and checking the MQDSS signature σ on it. For this last part, we use the latest MQDSS specification (2.1).

Evaluation of $\bar{\mathcal{P}}$. In several procedures, we have to evaluate the system $\bar{\mathcal{P}}(\mathbf{x}) = \mathcal{P}(\mathbf{x}_1)+\mathcal{R}(\mathbf{x}_2)$. We can use the special structure of the system to speed up the evaluation: we evaluate $\mathcal{P}$ and $\mathcal{R}$ separately and add the results afterwards. We use the technique described in [CKY21] to compute each system separately.

5.4 Benchmark results

We did a benchmark of our new post-quantum anonymous token scheme with the parameters and implementation details described above. The experiments were run on a 2017 MacBook Pro with a 2.3GHz Intel Core i5 processor and 8 GB of DDR3 RAM.

Key and token size. With the expression derived above, we obtain a private (resp. public) key of 237.88 KB (resp. 278.46 KB). The token length is 46.9 KB.

Performance. In our implementation, the computation of the $\mathcal{MQ}$ system is the most costly part, and some optimisation could be made to improve its efficiency. In fact, if we recall that 1 evaluation of $\mathcal{G}$ corresponds to 3 evaluations of the system $\bar{\mathcal{P}}$, the user evaluates the system once to check the answer from the issuer, $3r$ times when computing the commitments and r times in the computation process of the $\mathbf{e}_1^{(i)}$'s vectors, which gives a total of $4r+1$ times. About the verifier, it has to compute the system once if $b_i = 0$ and 4 times if $b_i = 1$, $i \in [r]$, which gives on average the verifier evaluating the system $2.5r$ times ($\Pr[b_i = 0] = \Pr[b_i = 1] = \frac{1}{2}$). We summarise the results in the Table 8, where we also report the metrics from the schemes discussed in Sections 3.2 and 3.3. The values in columns **user**, **issuer** and **verifier** are given in milliseconds.

scheme	tokens	λ	sig	trans	user	issuer	verifier
[PWFHW23] better size	1	115	85.6 kB	2.4 kB	4882	-	19.8
[PWFHW23] balanced	1	115	112,3 kB	2.4 kB	660	-	22
[PWFHW23] better time	1	115	173.3 kB	2.4 kB	304	-	31
[BLNS23] NTRU-ISIS$_f$	1	128	243 kB	473 kB	-	-	-
[BLNS23] Int-NTRU-ISIS$_f$	1	128	62 kB	107 kB	-	-	-
[Bas23]	1	128	32 B	8.7 MB	-	-	-
[ADDG23]*	1	128	32 B	4078,44 MB	256,1	123**	-
[ADDG23]* 64-queries amortised	1	128	2048 B	4078,41 MB	256,1	123**	-
[dSGP23]	1	λ	-	$(2\lambda + \frac{17}{2}) \log p + 4\lambda$	-	-	-
[PSM17]	1	128	28.5 kB	-	7760	19	5505
MQAT	1	128	46.9 kB	156 B	919	5	560
MQAT *modified*	100	128	15,63 kB	15.6 kB	50	500	1479***

Table 8: *Overall comparison.*
* *Implementation has not been made with verifiability so timings might be wrong.*
** *Estimated.*
*** *Verification time per token.*

Discussion. With the results obtained above, we can clearly see that our token and transcript sizes can compete with other state-of-the-art constructions. Moreover, the timings we obtain for the user, the issuer and the verifier are very promising. To get even more practical results, one can imagine the following modification in our anonymous token construction: stop the token issuance phase after the user checks the answer from the issuer. We end up with the issuance protocol presented in Figure 10. A token is then a pair $(t, (\mathbf{z}, \mathbf{z}^*))$ where t is a random tag in $\{0,1\}^{2\lambda}$ and $(\mathbf{z}, \mathbf{z}^*)$ is a solution to the system $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathsf{H}_m(t)$. In this case, the rest of the $\mathsf{User}_1(\cdot)$ procedure is performed during the redemption phase: the user first computes the MQDSS signature using the solution $(\mathbf{z}, \mathbf{z}^*)$ and then sends it to the verifier along with the tag t to be checked. With this construction, the user computes the overall system $\bar{\mathcal{P}}$ only once during the issuance phase, which allows for batching token requests and even more practical results. By extrapolating the numbers obtained above and using the modified scheme, a user could receive 100 tokens in $\sim$ 550 ms and verify each of them in $\sim$ 1470 ms; the signature size would also decrease as the MQDSS signature would only be computed later when a token is spent. We summarize these metrics under MQAT *modified* in Table 8. Comparing with the other schemes, isogeny-based constructions achieve the most compact signature sizes, but can be extremely slow in the verification procedure. Also, their transcript sizes are impractical, and even more in the context of high-latency networks. In a setting where the network has a low bandwidth, such as assumed in a shared IP setting, the numbers obtained using MQAT seem very reasonable and offer in our opinion a better trade-off between sizes and performance. MQAT has also more compact signature and transcript sizes than [PWFHW23] and [BLNS23]. The performance of [PWFHW23] is comparable with ours but again the trade-off between signature size, transcript size, performance and practicality seems in the favor of MQAT.

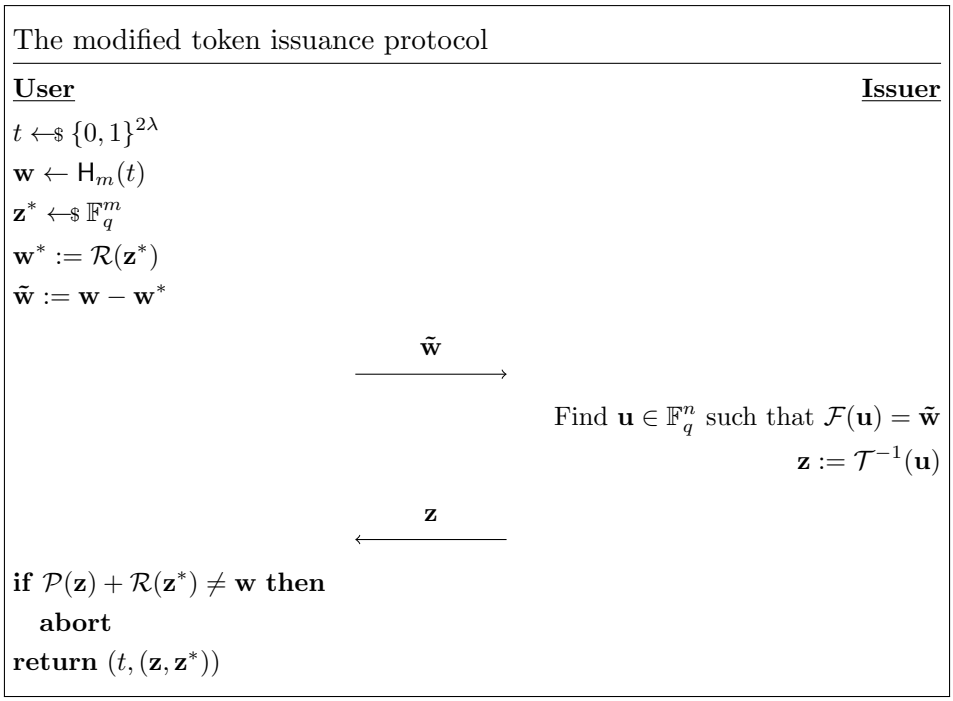

Figure 10: *The modified issuance protocol for* MQAT.

6 Conclusion

In conclusion, this work has provided a comprehensive exploration of anonymous tokens, ranging from classical cryptography constructions to innovative schemes rooted in post-quantum cryptography. By delving into the principles of various cryptographic primitives, including post-quantum VOPRFs, blind signatures, and multivariate cryptography, we have contributed to the evolving landscape of privacy-preserving technologies. Our novel post-quantum anonymous tokens construction, based on multivariate cryptography, fulfills the usual security properties of *unlinkability* and *one-more token unforgeability* but also adds post-quantum resilience. We also stress that our scheme is publicly verifiable. The proposed instantiation achieves a quantum security of 128 bits. Notably, the protocol's efficiency, evident in short token and transcript sizes, positions it competitively against contemporary constructions. Furthermore, the potential for optimizing the scheme, as evidenced by the prospect of batch issuance, adds a layer of practicality to its theoretical merits. Looking ahead, future endeavors could explore the integration of private and/or public metadata into tokens, offering an avenue for enhancing customisation within the framework of anonymous tokens. This work stands as a testament to the ongoing quest for robust, efficient, and privacy-centric cryptographic solutions in the face of emerging challenges.

References

[ADDG23] Martin R. Albrecht, Alex Davidson, Amit Deo, and Daniel Gardham. Crypto Dark Matter on the Torus: Oblivious PRFs from shallow PRFs and FHE. Cryptology ePrint Archive, Paper 2023/232, 2023. https://eprint.iacr.org/2023/232.

[Bas23] Andrea Basso. A Post-Quantum Round-Optimal Oblivious PRF from Isogenies. Cryptology ePrint Archive, Paper 2023/225, 2023. https://eprint.iacr.org/2023/225.

[BCC+10] Charles Bouillaguet, Hsieh-Chung Chen, Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, Adi Shamir, and Bo-Yin Yang. Fast Exhaustive Search for Polynomial Systems in $\mathbb{F}_2$. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010*, Lecture Notes in Computer Science, pages 203–218, Berlin, Heidelberg, 2010. Springer.

[BCH+23] Ward Beullens, Ming-Shing Chen, Shih-Hao Hung, Matthias J. Kannwischer, Bo-Yuan Peng, Cheng-Jhih Shih, and Bo-Yin Yang. Oil and Vinegar: Modern Parameters and Implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(3):321–365, June 2023.

[Beu22] Ward Beullens. Breaking Rainbow Takes a Weekend on a Laptop. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, Lecture Notes in Computer Science, pages 464–479, Cham, 2022. Springer Nature Switzerland.

[BFP09] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, September 2009. Publisher: De Gruyter.

[BKW20] Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious Pseudorandom Functions from Isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, Lecture Notes in Computer Science, pages 520–550, Cham, 2020. Springer International Publishing.

[BLNS23] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Alessandro Sorniotti. A Framework for Practical Anonymous Credentials from Lattices. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, Lecture Notes in Computer Science, pages 384–417, Cham, 2023. Springer Nature Switzerland.

[BLOR22] Fabrice Benhamouda, Tancrède Lepoint, Michele Orrù, and Mariana Raykova. Publicly verifiable anonymous tokens with private metadata bit. Cryptology ePrint Archive, Paper 2022/004, 2022. https://eprint.iacr.org/2022/004.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, Lecture Notes in Computer Science, pages 514–532, Berlin, Heidelberg, 2001. Springer.

[CD23] Wouter Castryck and Thomas Decru. An Efficient Key Recovery Attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, Lecture Notes in Computer Science, pages 423–447, Cham, 2023. Springer Nature Switzerland.

[CDV23] Melissa Chase, F. Betül Durak, and Serge Vaudenay. Anonymous Tokens with Stronger Metadata Bit Hiding from Algebraic MACs. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, Lecture Notes in Computer Science, pages 418–449, Cham, 2023. Springer Nature Switzerland.

[CHR+16] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-Pass $\mathcal{MQ}$-Based Identification to $\mathcal{MQ}$-Based Signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, Lecture Notes in Computer Science, pages 135–165, Berlin, Heidelberg, 2016. Springer.

[CKY21] Tung Chou, Matthias J. Kannwischer, and Bo-Yin Yang. Rainbow on Cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 650–675, August 2021.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An Efficient Post-Quantum Commutative Group Action. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, Lecture Notes in Computer Science, pages 395–427, Cham, 2018. Springer International Publishing.

[CMZ14] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. Algebraic MACs and Keyed-Verification Anonymous Credentials. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 1205–1216, New York, NY, USA, November 2014. Association for Computing Machinery.

[Cou06] Jean-Marc Couveignes. Hard Homogeneous Spaces. Cryptology ePrint Archive, Paper 2006/291, 2006. https://eprint.iacr.org/2006/291.

[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.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, February 2018.

[DKPW12] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. Message Authentication, Revisited. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, Lecture Notes in Computer Science, pages 355–374, Berlin, Heidelberg, 2012. Springer.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, Lecture Notes in Computer Science, pages 164–175, Berlin, Heidelberg, 2005. Springer.

[dSGP23] Cyprien Delpech de Saint Guilhem and Robi Pedersen. New proof systems and an OPRF from CSIDH. Cryptology ePrint Archive, Paper 2023/1614, 2023. https://eprint.iacr.org/2023/1614.

[Fau02] Jean Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, ISSAC '02, pages 75–83, New York, NY, USA, July 2002. Association for Computing Machinery.

[FMP23] Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-SIDH and MD-SIDH: Countering SIDH Attacks by Masking Information. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, Lecture Notes in Computer Science, pages 282–309, Cham, 2023. Springer Nature Switzerland.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, July 1996. Association for Computing Machinery.

[JDF11] David Jao and Luca De Feo. Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, Lecture Notes in Computer Science, pages 19–34, Berlin, Heidelberg, 2011. Springer.

[JV18] Antoine Joux and Vanessa Vitse. A Crossbred Algorithm for Solving Boolean Polynomial Systems. In Jerzy Kaczorowski, Josef Pieprzyk, and Jacek Pomykała, editors, *Number-Theoretic Methods in Cryptology*, Lecture Notes in Computer Science, pages 3–21, Cham, 2018. Springer International Publishing.

[KLOR20] 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*, Lecture Notes in Computer Science, pages 308–336, Cham, 2020. Springer International Publishing.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, Lecture Notes in Computer Science, pages 206–222, Berlin, Heidelberg, 1999. Springer.

[KS98] Aviad Kipnis and Adi Shamir. Cryptanalysis of the oil and vinegar signature scheme. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, Lecture Notes in Computer Science, pages 257–266, Berlin, Heidelberg, 1998. Springer.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A Direct Key Recovery Attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, Lecture Notes in Computer Science, pages 448–471, Cham, 2023. Springer Nature Switzerland.

[Pat97] Jacques Patarin. The oil and vinegar signature scheme. In *Presented at the Dagstuhl Workshop on Cryptography September 1997*, 1997.

[PSM17] Albrecht Petzoldt, Alan Szepieniec, and Mohamed Saied Emam Mohamed. A Practical Multivariate Blind Signature Scheme. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 437–454, Cham, 2017. Springer International Publishing.

[PWFHW23] Guru-Vamsi Policharla, Bas Westerbaan, Armando Faz-Hernández, and Christopher A Wood. Post-Quantum Privacy Pass via Post-Quantum Anonymous Credentials. Cryptology ePrint Archive, Paper 2023/414, 2023. https://eprint.iacr.org/2023/414.

[Rob23] Damien Robert. Breaking SIDH in Polynomial Time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, Lecture Notes in Computer Science, pages 472–503, Cham, 2023. Springer Nature Switzerland.

[RS06] Alexander Rostovtsev and Anton Stolbunov. PUBLIC-KEY CRYPTOSYSTEM BASED ON ISOGENIES. Cryptology ePrint Archive, Paper 2006/145, 2006. https://eprint.iacr.org/2006/145.

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, November 1994.

[SS22] Tjerand Silde and Martin Strand. Anonymous Tokens with Public Metadata and Applications to Private Contact Tracing. In Ittay Eyal and Juan Garay, editors, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 179–199, Cham, 2022. Springer International Publishing.

[SSH11] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, Lecture Notes in Computer Science, pages 706–723, Berlin, Heidelberg, 2011. Springer.

[SW16] Peter Schwabe and Bas Westerbaan. Solving Binary $\mathcal{MQ}$ with Grover's Algorithm. In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering*, Lecture Notes in Computer Science, pages 303–322, Cham, 2016. Springer International Publishing.

[YC05] Bo-Yin Yang and Jiun-Ming Chen. All in the XL Family: Theory and Practice. In Choon-sik Park and Seongtaek Chee, editors, *Information Security and Cryptology – ICISC 2004*, Lecture Notes in Computer Science, pages 67–86, Berlin, Heidelberg, 2005. Springer.

A Code for finding the number of MQDSS rounds

```
import math

k = 128
q = 256.0
r_min = 128
r_max = 256
one_over_q = 1/q
q_minus_one_over_q = (q-1)/q
goal = 2 ** 128
one_over_goal = 1.0/goal

def pr(N, r):
    sum = 0.0
    for i in range(N, r):
        a = one_over_q ** i
        b = q_minus_one_over_q ** (r-i)
        c = math.comb(r, i)
        tmp = a * b * c
        sum += tmp
    return sum

if __name__ == "__main__":
    for r in range(r_min, r_max):
        total = 0
        for i in range(r):
            pri = pr(i, r)
            if pri <= one_over_goal:
                total += 1
                continue

            one_over_pri = 1/pri
            two_power_r_minus_i = pow(2, r-i)
            tmp = one_over_pri + two_power_r_minus_i
            if tmp >= goal:
                total += 1
        if total == r:
            print(f"The number of round is {r}")
            break
```