On the Efficiency of Revocation in RSA-Based Anonymous Systems

María Fueyo and Javier Herranz

Dept. Matemàtiques,
Universitat Politècnica de Catalunya,
C. Jordi Girona 1-3, Mòdul C3, 08034, Barcelona, Spain.
e-mail: mariafueyo@gmail.com , javier.herranz@upc.edu

Abstract. The problem of revocation in anonymous authentication systems is subtle and has motivated a lot of work. One of the preferable solutions consists in maintaining either a whitelist $\mathcal{L}_W$ of non-revoked users or a blacklist $\mathcal{L}_B$ of revoked users, and then requiring users to additionally prove, when authenticating themselves, that they are in $\mathcal{L}_W$ (membership proof) or that they are not in $\mathcal{L}_B$ (non-membership proof). Of course, these additional proofs must not break the anonymity properties of the system, so they must be zero-knowledge proofs, revealing nothing about the identity of the users.

In this paper we focus on the RSA-based setting and we consider the case of non-membership proofs to blacklists $\mathcal{L} = \mathcal{L}_B$. The existing solutions for this setting rely on the use of universal dynamic accumulators [20]; the underlying zero-knowledge proofs are a bit complicated and thus their efficiency, although being independent from the size of the blacklist $\mathcal{L}$, seems to be improvable. Peng and Bao [21] already tried to propose simpler and more efficient zero-knowledge proofs for this setting, but we prove in this paper that their protocol is not secure. We fix the problem by designing a new protocol and formally proving its security properties. We then compare the efficiency of the new zero-knowledge non-membership protocol with that of the protocol in [20], when they are integrated with anonymous authentication systems based on RSA (notably, the IBM product Idemix for anonymous credentials). We discuss for which values of the size k of the blacklist $\mathcal{L}$ one protocol is preferable to the other one and we propose different ways to combine and implement the two protocols.

Keywords: anonymous authentication, RSA, revocation, non-membership arguments.

1 Introduction

1.1 The Problem

This paper deals with real-life systems providing anonymous authentication and supporting revocation of users. The specific example that we have in mind is the IBM system for anonymous credentials, Identity Mixer (Idemix, for short) [18].

Anonymous Authentication. In several digital scenarios, a user has to authenticate himself in a somewhat private or anonymous way. For instance, in order to gain access to a certain resource, or in order to sign a digital document, the user has to prove that he is a member of a certain group of users, or that he enjoys enough rights or attributes, or that he has explicit permission (a credential) to do so, without revealing his identity or the specific rights, attributes or credentials that he holds. Three canonical examples of such real-life situations are: anonymous, but restricted, access control to buildings or digital resources; restricted but anonymous signed opinions in digital polls; anonymous but reliable leakage of information to the press.

The cryptographic community has defined different primitives for these cases, including anonymous credentials, group, ring and attribute-based signatures, or direct anonymous attestation, to name a few. There are specific constructions of these primitives in different mathematical settings: the RSA-based one, the pairing-based one, the lattice-based one, etc. In this work we will focus on those solutions that work in the RSA-based setting, like Idemix.

Incorporating Revocation of Users. A feature which is very related to anonymous authentication is that of revocation. When a user misbehaves, or when a user looses his attributes (for instance, if he leaves a company where the system is being implemented), then this user should not be able to successfully run the anonymous authentication process anymore. In other words, the user should be revoked from the system. Combining revocation with anonymity is not trivial at all, because the solution of publishing the identities of the revoked users and then letting users reveal their identity when running authentication is obviously not anonymous. There are different solutions proposed in the literature for this problem, and one of the most efficient ones (as concluded in [19], where the specific case of Idemix is considered) is the use of revocation (black)lists: each user U_i is related (publicly or privately) to a value e_i. When a user is revoked from the system, his value e_i is added to a revocation list $\mathcal{L}$. The system administrator publishes some information related to $\mathcal{L}$, which can eventually be the list itself. Then, in the authentication process, a user U with value e must additionally prove in an anonymous way that his value is not in the list $\mathcal{L}$.

Non-Membership Zero-Knowledge Arguments. Therefore, the problem of adding revocation to anonymous authentication systems can be reduced to the problem of designing a protocol to prove that a value e committed in C_e: (i) is consistent with the value of e hidden in the credential of the user, and (ii) does not belong to the revocation list $\mathcal{L}$. Such a protocol must reveal nothing else on e. This kind of protocols are known as *non-membership zero-knowledge arguments.*

There exist different proposals of such protocols, in different mathematical settings, and with different efficiency properties (in particular, regarding the dependence on the cardinality k of the blacklist $\mathcal{L}$). For instance, in the traditional discrete logarithm setting, there are different solutions [21, 4, 2], where the size of the actual non-membership proof is $\mathcal{O}(k), \mathcal{O}(\sqrt{k}), \mathcal{O}(\log k)$, respectively. In other settings, the primitive of universal and dynamic cryptographic accumulators [9, 20] can be used to obtain non-membership proofs whose size is independent of k. This is possible in the pairing-based setting (by using the accumulator-based protocols in [6, 11]) and in the RSA-based setting (by using the accumulator-based protocol in [20]).

In this paper we focus on the RSA-based setting. Therein, the best known solution for non-membership zero-knowledge proofs is the protocol in [20], which is quite complex and relatively inefficient, although producing constant-size proofs (independent of the number k of elements in the blacklist $\mathcal{L}$). Let us sketch the basic idea underlying the solution in [20]: elements in the blacklist $\mathcal{L} = \{e_1, \ldots, e_k\}$ are prime numbers bigger than 1; therefore, an integer element e is not in $\mathcal{L}$ if and only if it is coprime with the product $\prod_{i=1}^{k} e_i$. By Bezout's identity, this condition is equivalent to the existence of integer numbers $a, b \in \mathbb{Z}$ such that $ae + b \prod_{i=1}^{k} e_i = 1$. Therefore, such a pair (a, b) of integer elements is a secret witness of the fact that $e \notin \mathcal{L}$, and could be used to prove this fact in a zero-knowledge way, without revealing anything about e, a, b. The problem with this approach is that the size of the witness pair (a, b), and later the efficiency and length of the zero-knowledge proof, depend on the size of the blacklist $\mathcal{L}$, because the solutions a, b to the Bezout's equality $ae + b \prod_{i=1}^{k} e_i = 1$ satisfy, in general, $|a| \leq \prod_{i=1}^{k} e_i$ and $|b| \leq e$. Thus, if we assume that all the elements in $\mathcal{L}$ are k_e bits long, then the integer a will be $k \cdot k_e$ bits long, in general. Since the goal in [20] was to provide constant-size zero-knowledge arguments, independently of the size k of $\mathcal{L}$, they replaced the witness (a, b) with a witness pair (g^a, b), where $\langle g \rangle$ is a cyclic group. In this way, the size of the witness and the cost of computing and sending the zero-knowledge proofs become independent on k; but the resulting zero-knowledge protocols are quite complicated and inefficient, because proving the knowledge of g^a satisfying some properties turns to be more involved than proving the knowledge of an integer a.

A natural question is, then: what happens if we keep the pair (a, b) as the secret witness, and we design zero-knowledge protocols that use this witness? The efficiency of these protocols and the length of the resulting proofs will depend on $k = |\mathcal{L}|$, but maybe the protocols become simpler and the resulting proofs are still shorter than those produced by the protocol in [20], for some values of k. This is precisely what Peng and Bao explored in the second part of their paper [21]: they proposed a protocol to prove non-membership in zero-knowledge, for the specific RSA-based setting where elements in $\mathcal{L}$ satisfy some coprimality conditions, using the pair of integers (a, b) as the secret witness. This protocol is much simpler than the protocol in [20].

1.2 Our Contributions

The first contribution of this paper is the description of an attack against the aforementioned protocol of Peng and Bao [21] for a zero-knowledge non-membership proof in the RSA-based setting. Namely, a user whose value $e \in \mathcal{L}$ is included in the list, can still produce a non-membership proof that is admitted by the verifier(s) with significant probability. A bit more technically, this means that the protocol by Peng and Bao does not satisfy the soundness property; actually, no formal security proof for this property was given in [21].

Then, we fix this security problem by proposing a new protocol for non-membership in that RSA-based setting where the elements in the list $\mathcal{L}$ are all different prime numbers, and where the secret witness for the prover is the pair of integers (a, b) satisfying $ae + b \prod_{i=1}^{k} e_i = 1$. The resulting protocol is less efficient than the insecure protocol of Peng and Bao, but we formally prove that it satisfies the required properties for this kind of arguments: correctness, zero-knowledge and soundness.

Exactly as it happens with the protocol in [20], our new protocol can be combined with some existing anonymous authentication systems in the RSA-based setting [1, 17, 7, 7, 18, 5] so that they enjoy the feature of revocation. We thus provide an efficiency comparison between our new zero-knowledge non-membership arguments and those in [20]: the conclusion is that the new protocol is more efficient for small blacklists. We discuss different ways of implementing these two protocols, in independent or combined ways, along with the advantages and drawbacks of each choice. We believe the new protocol and the detailed comparison with the protocols in [20] lead to an improvement in both the understanding and the state of the art of RSA-based anonymous authentication systems with revocation, in particular in Idemix.

2 Mathematical Setting and Building Blocks

In our scheme, we will consider the following probabilistic algorithm, $\mathsf{RSA.Inst}$, to generate the necessary mathematical parameters. $\mathsf{RSA.Inst}$ takes as input a security parameter $\lambda \in \mathbb{Z}^+$, and then chooses two random prime numbers, P, Q, each one being $\lambda/2$-bits long, such that both $p = \frac{P-1}{2}$ and $q = \frac{Q-1}{2}$ are also prime. The public modulus will be $N = PQ$. Let $QR(N) = \{z^2 \bmod N \mid z \in \mathbb{Z}_N^*\} \subset \mathbb{Z}_N^*$ be the set of quadratic residues modulo N. $QR(N)$ is a cyclic group of order pq; the algorithm $\mathsf{RSA.Inst}$ generates at random an element $g \in QR(N)$ with order q, for instance by taking at random an element $w \in \mathbb{Z}_N^*$ and defining $g = w^{2p} \bmod N$. We denote an execution of this algorithm as $(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$. We will use such RSA moduli, product of safe primes, for simplicity; the results of this work are valid for more general RSA moduli, those described in [14].

The security analysis of our scheme will be based on the hardness of the following mathematical problem in $QR(N)$, introduced in [13].

Definition 1. (Strong QR-RSA Problem [13].) *An algorithm $\mathcal{A}_{sRSA}$ solves the Strong QR-RSA problem if it receives as input N and a randomly chosen $\omega \overset{R}{\leftarrow} QR(N)$, where $(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$, and outputs $\phi \in \mathbb{Z}_N^*$ and an integer $e > 1$ such that $\omega = \phi^e \bmod N$.*

The Strong QR-RSA Assumption states that the probability that any algorithm $\mathcal{A}_{sRSA}$ solves the Strong QR-RSA problem in polynomial time is negligible in λ, meaning that this probability decreases (as λ increases) faster than the inverse of any polynomial. Formally, for any algorithm $\mathcal{A}_{sRSA}$ running in polynomial time, we have that the function

$$\Pr\left[\begin{array}{l|l} (\phi, e) \leftarrow \mathcal{A}_{sRSA}(N, \omega), & (P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda); \\ e > 1,\ \omega = \phi^e \bmod N & \omega \overset{R}{\leftarrow} QR(N); \end{array}\right](\lambda)$$

is negligible in λ.

Under the assumption that factoring an RSA modulus is hard and if N is the product of two safe primes, the Strong QR-RSA Assumption is equivalent (see for instance [13, 12]) to the (more commonly used) Strong RSA Assumption, related to the Strong RSA problem, which only differs to the Strong QR-RSA problem in the fact that $\omega \overset{R}{\leftarrow} \mathbb{Z}_N^*$ instead of $\omega \overset{R}{\leftarrow} QR(N)$.

A specific result that will be used in the security proofs for the new non-membership zero-knowledge protocol, in Section 5, is stated and proved below.

Lemma 1. *Under the Strong QR-RSA Assumption, if an adversary $\mathcal{F}$, given as inputs N and $g \in QR(N)$, is able to obtain values $L \in QR(N)$, $a, v \in \mathbb{Z} - \{0\}$ such that $L^a = g^v \bmod N$, then it must hold $\frac{v}{a} \in \mathbb{Z}$.*

Proof. Let $d = \gcd(v, a)$ be the largest integer dividing both v and a. This means (Euclidean algorithm) that there exist integers $\rho_a, \rho_v \in \mathbb{Z}$ such that $\rho_a a + \rho_v v = d$. Let us assume, for the sake of contradiction, that $\frac{v}{a} \notin \mathbb{Z}$. Therefore, it must hold $a > d$ and, so, $e := \frac{a}{d} > 1$.

Now we have that $g = g^{\frac{\rho_a a + \rho_v v}{d}} = g^{\rho_a e} L^{\rho_v e} = (g^{\rho_a} L^{\rho_v})^e \bmod N$. Therefore, running $\mathcal{F}$ with inputs N and $g = \omega$ would solve the instance (N, ω) of the Strong RSA problem, because $g^{\rho_a} L^{\rho_v}$ is an e-th root of g, for $e = \frac{a}{d} > 1$. This would contradict the Strong QR-RSA Assumption, and so we can conclude that $\frac{v}{a} \in \mathbb{Z}$. □

2.1 The Damgård-Fujisaki Commitment Scheme

A commitment scheme Cmt is defined by three protocols ($\mathsf{Cmt.Setup}$, $\mathsf{Cmt.Commit}$, $\mathsf{Cmt.Check}$). The first protocol takes as input a security parameter and outputs some common public parameters (or reference string). The second protocol takes as input the value x to be committed and randomness r, and outputs the commitment $C_x = \mathsf{Cmt.Commit}(x, r)$. Finally, given a commitment C, a value x and randomness r, the protocol $\mathsf{Cmt.Check}(C, x, r)$ simply consists in verifying the equation $C = \mathsf{Cmt.Commit}(x, r)$.

A commitment scheme must satisfy two properties: the hiding and binding properties. The first one informally means that C_x does not reveal anything about x. The second one means that it must be hard to find two valid and different openings for the same commitment; that is, to find (x_1, r_1, x_2, r_2, C) such that $x_1 \neq x_2$ and both $\mathsf{Cmt.Check}(C, x_1, r_1)$ and $\mathsf{Cmt.Check}(C, x_2, r_2)$ return true.

Damgård and Fujisaki proposed in [14] a commitment scheme for integer values, which is an extension of the Fujisaki-Okamoto commitment scheme [16]. Using the previous notation, the common public parameters of the Damgård-Fujisaki commitment scheme are (N, g, h, γ), where $(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$, $h \overset{R}{\leftarrow} QR(N)$ (for instance, by taking $\alpha \overset{R}{\leftarrow} \mathbb{Z}_N$ and defining $h = g^\alpha \bmod N$) and γ is such that $pq = ord(QR(N)) \leq 2^\gamma$; since $N = PQ \approx 2^\lambda$, the value $\gamma = \lambda - 2$ is enough. The commitment to an integer element e is defined as $C_e = g^e h^r \bmod N$, where $r \overset{R}{\leftarrow} \mathbb{Z}_{2^{\gamma+\lambda}}$ is the randomness of the protocol.

The hiding property of this protocol holds statistically, whereas the binding property holds computationally, assuming the hardness of the factoring problem: if an adversary breaks the binding property, then it can be used to factor N.

3 Non-Membership Arguments: Protocols and Security

The main topic of this work are non-membership zero-knowledge arguments of knowledge; these are interactive protocols where a prover P tries to convince a verifier V of the fact that P knows an element e which is committed in some public value C_e and which is not included in a public list $\mathcal{L} = \{e_1, \ldots, e_k\}$ of elements.

The definitions (protocols and properties) for zero-knowledge arguments of knowledge, for a general language, can be found in many works in the literature. In order to be more direct and specific, we will write the definitions for the particular case that we consider in this work: non-membership to a public list $\mathcal{L}$.

- A setup algorithm $\mathcal{G}$ takes as input a security parameter λ and outputs a common reference string crs, in our case the description of the parameters (N, g, h) for the commitment scheme, and the list $\mathcal{L} = \{e_1, \ldots, e_k\}$ of revoked elements.
- Later, the interaction between the prover P and the verifier V consists of an exchange of messages between them. The initial input for the prover P is crs and elements e, r, whereas the initial input for the verifier is crs and a commitment C_e. The final output for the verifier is a bit $b \in \{0, 1\}$, which is $b = 0$ if he rejects and $b = 1$ if he accepts the proof. The public transcript of such an interaction is denoted as $tr \leftarrow \langle P(\mathsf{crs}, e, r), V(\mathsf{crs}, C_e)\rangle$, whereas an execution of the interactive protocol is denoted as $\langle P(\mathsf{crs}, e, r), V(\mathsf{crs}, C_e)\rangle = b$.

The size of a proof will be the size of the transcript tr.

3.1 Security Definitions

Such a protocol is called a zero-knowledge and extractable non-membership argument if it satisfies the three properties of perfect completeness, honest verifier zero-knowledge and computational soundness, as defined as follows.

Completeness. If both the prover and the verifier behave honestly and the prover actually knows the values e, r which satisfies the claimed statement, then the proof should always be accepted as valid. That is, for all execution $\mathcal{G}(1^\lambda) \rightarrow \mathsf{crs} = (N, g, h, \mathcal{L})$ with $\mathcal{L} = \{e_1, \ldots, e_k\}$, and all values $e \notin \mathcal{L}$, $r \in \mathbb{Z}_N$, then $\langle P(\mathsf{crs}, e, r), V(\mathsf{crs}, C_e)\rangle = 1$, if $C_e = g^e h^r \bmod N$.

Soundness. The soundness property reflects the intuition that a dishonest prover cannot succeed in convincing a verifier about a false statement. In other words, if an execution of the proof produces $b = 1$ as the output for the verifier, then it must be the case that the prover knows values e, r such that $e \notin \mathcal{L}$ and $c_e = g^e h^r \bmod N$.

The way of formalizing this property is by requiring, for any deterministic polynomial time prover P^*, the existence of a probabilistic polynomial-time extractor algorithm E such that: for any execution $\mathcal{G}(1^\lambda) \rightarrow \mathsf{crs} = (N, g, h, \mathcal{L})$ and for any execution of the interactive protocol, if $\varepsilon = \Pr[\langle P^*(\mathsf{crs}, e^*, r^*), V(\mathsf{crs}, C_e)\rangle = 1]$, then the probability that E, given inputs (crs, C_e) and oracle access to P^* (playing E the verifier role), outputs e, r such that $e \notin L$ and $C_e = g^e h^r \bmod N$ is polynomially related to ε.

Zero-Knowledge. Intuitively, this property ensures that an execution of the protocol does not leak any information about the secret witness e used by the prover, other than the fact that $e \notin \mathcal{L}$ (if the proof is accepted), assuming that the verifier runs his part of the protocol honestly. A stronger version of the definition can be considered, for dishonest verifiers V^*, but in this work the weaker notion will suffice.

Honest verifier zero-knowledge holds if there exists a probabilistic polynomial-time simulator algorithm S such that: for any common reference string $\mathcal{G}(1^\lambda) \to \mathsf{crs} = (N, g, h, \mathcal{L})$ and for any value $C_e = g^e h^r \bmod N$, $\mathsf{S}(\mathsf{crs}, C_e)$ generates a transcript tr_{sim} whose distribution is (computationally) indistinguishable from that of transcripts $tr = \langle P(\mathsf{crs}, e, r), V(\mathsf{crs}, C_e)\rangle$ generated by honestly following the protocol.

4 Insecurity of a Protocol by Peng and Bao

We first describe the protocol proposed by Peng and Bao for special applications (in Section 3 of [21]), and then we explain a specific attack against it.

4.1 Description of the Protocol

In this protocol, there is a public list of integer values, $\mathcal{L} = \{e_1, e_2, \ldots, e_k\}$, and a prover has committed an integer element e into a commitment C_e. This prover wants to prove, in zero-knowledge, that the committed value e is not in the list $\mathcal{L}$. The commitment algorithm is the one by Damgård-Fujisaki [14]: the value e is committed in $C_e = g^e h^r \bmod N$ where r is randomly chosen in $\mathbb{Z}_{2^\gamma N}$. The protocol of Peng and Bao (in Section 3 of [21]) works in the special case where e is loosely coprime with the elements in $\mathcal{L}$; that is, $\gcd(e, \prod_{i=1}^k e_i) < \min(|e_1|, |e_2|, \ldots, |e_k|)$.

So, once again, the common reference string of this protocol can be thought as $\mathsf{crs} = (N, g, h, \gamma, \mathcal{L}) \leftarrow \mathcal{G}(1^\lambda)$. Later, the interaction between the prover P with inputs $(\mathsf{crs}, e, r, C_e)$ and verifier V with inputs (crs, C_e) works as follows.

- Both P and V compute $C = g^{e_1 e_2 \cdots e_k} \bmod N$.
- If $j = \gcd(e, \prod_{i=1}^k e_i)$, then P uses the Euclidean algorithm to compute integers $a, b \in \mathbb{Z}$ satisfying $ae + b\prod_{i=1}^k e_i = j$.
- Now, defining $s = \min(|e_1|, |e_2|, \ldots, |e_n|)$, $z = ar$, an interative protocol is run by P and V. The steps of this protocol are the following ones:
  1. P publishes $F = g^j h^{r'} \bmod N$, where r' is randomly chosen from $\mathbb{Z}_N$.
  2. P proves that the integer j committed in F is in the range $\{1, 2, \cdots, s-1\}$, using the proof in [3].
  3. In parallel, P proves that he knows secret integers a, b, j, z, r' such that

$$C_e^a \cdot C^b = g^j h^z \bmod N \quad \text{and} \quad F = g^j h^{r'} \bmod N,$$

     through the following protocol (where ρ, τ are big enough integer values, specified by a security parameter; for instance, their bit length may be four times bigger than that of N):

     (3.1) P randomly chooses integers $u, v, w' \in \mathbb{Z}_\rho$, computes the values

$$y = ue + v\prod_{i=1}^{k} e_i \in \mathbb{Z}, \quad w = ur \in \mathbb{Z},$$

$$A = g^y h^{w'} \bmod N \in \mathbb{Z}_N$$

     and sends A to the verifier V.

     (3.2) V chooses a random challenge $c \in \mathbb{Z}_\tau$ and sends it to P.

     (3.3) P computes the following 5 values in $\mathbb{Z}$ and publishes them:

$$b_1 = u - ca, \quad b_2 = v - cb, \quad b_3 = y - cj,$$

$$b_4 = w - cz, \quad b_5 = w' - cr'.$$

(3.4) The verifier V checks that these two equations hold:

$$C_e^{b_1} C^{b_2} = g^{b_3} h^{b_4} \bmod N$$

$$g^{b_3} h^{b_5} F^c = A \bmod N$$

The final output of the verifier V is $b = 1$ only if both the proof in Step 2 and the checking in Step (3.4) are accepted.

4.2 Description of an Attack

A dishonest prover P^* whose secret value e, committed as $C_e = g^e h^r \bmod N$, belongs to the list $\mathcal{L}$, is still able to fool the verifier with a probability of $1/e$, through the following procedure:

1. Define $j = 1$, $a = \frac{1}{e}$ (a rational number), $b = 0$, and run all the steps of the Peng-Bao protocol, until Step (3.1).
2. In Step (3.2), receive the challenge $c \in \mathbb{Z}_\tau$ from the verifier V.
3. If c is not a multiple of e, abort. Otherwise, if e divides c, which happens with probability $1/e$, compute elements $b_1, \ldots, b_5 \in \mathbb{Z}$ as in Step (3.3) of the protocol.

With probability $1/e$, the random challenge c is a multiple of e and so all the values $b_1, \ldots, b_5$ are integers, and the proof is accepted as valid because all the equations checked by the verifier are satisfied. Therefore, the dishonest prover P^* convinces the verifier even if the claimed statement is not true, because $e \in \mathcal{L}$.

In other words, the protocol by Peng-Bao for special applications (described in Section 3 of [21]) does not enjoy the soundness property. A possibility to avoid this attack is to consider only very large values e for the users of the authentication system, so that the success probability of the attack becomes negligible. However, even in this case it would not be clear at all how to formally prove the soundness property of the Peng-Bao protocol: in a nutshell, the first verification equation (step 3.4) does not depend on the challenge c; this means that the secret witnesses which only appear in that first equation (in particular, integers a, b which appear inside b_1, b_2) could not be extracted in standard proof of the soundness security property, based on rewinding an execution of a successful prover P^*.

Taking this last observation in mind, in the next section we propose a way to fix the Peng-Bao protocol in order to obtain an efficient non-membership zero-knowledge argument for special applications (in the RSA-based setting), with provable security.

5 A New Non-Membership Zero-Knowledge Argument

For simplicity, and since this is the case in many RSA-based anonymous authentication systems (in particular, Idemix [18]), we will assume that the elements in the revocation list $\mathcal{L}$ are prime numbers. Therefore, a first modification to the Peng-Bao protocol described in the previous section is that, for elements $e \notin \mathcal{L}$, it will always be the case that e is coprime with any single element in $\mathcal{L}$, and so $j = 1$ will always hold. In particular, this will mean that the range proof for the (secret) value j in the Peng-Bao protocol (Step 2) is not needed anymore.

5.1 Description of the New Protocol

Setup, $\mathcal{G}(1^\lambda) \to \mathsf{crs}$. Given a security parameter λ, run $(P, Q, N, g) \leftarrow \mathsf{RSA.Inst}(1^\lambda)$ to instantiate the RSA setting. Again, the used commitment scheme will be that of Damgård-Fujisaki. Therefore, an additional element $h \in QR(N)$ is chosen by sampling $\alpha \overset{R}{\leftarrow} \mathbb{Z}_{2^{2\gamma+\lambda}}$ and computing $h = g^\alpha \bmod N$, where $\gamma = \lambda - 2$. Let $\mathcal{L} = \{e_1, \ldots, e_k\}$ be a list containing k different prime numbers, and let us

assume that $e_i < 2^{k_e}$, for all $e_i \in \mathcal{L}$, for some upper bound $k_e \geq \lambda$. Choose an additional security parameter $\kappa \in \mathbb{N}$ satisfying $2^\kappa \leq q$, where $Q = 2q + 1$.
Define $\mathsf{crs} = (N, g, h, \gamma, k_e, \kappa, \mathcal{L})$.

Interaction, $tr \leftarrow \langle P(\mathsf{crs}, e, r), V(\mathsf{crs}, C_e)\rangle$**.** Suppose P knows integer values e, r such that $e \notin \mathcal{L}$ and $C_e = g^e h^r \bmod N$. We assume $0 < e < 2^{k_e}$ and $r \in \mathbb{Z}_{2^{\gamma+\lambda}}$. The interactive protocol between P and V works as follows:

1. Both P and V compute $C = g^{e_1 e_2 \ldots e_k} \bmod N$.
2. P uses the Euclidean algorithm to compute integers $a, b \in \mathbb{Z}$ satisfying
$$ae + b\prod_{i=1}^{k} e_i = 1,$$
such that $|a| \leq \prod_{i=1}^{k} e_i \leq \left(2^{k_e}\right)^k$ and $|b| \leq e < 2^{k_e}$.
3. P computes the integer value $z = ar$.
4. Now P and V are ready to run the 3-moves interaction itself, which using the terminology in [10], is a zero-knowledge proof of knowledge ZKPK$\{(e, r, a, b, z)$ s.t. $C_e = g^e h^r \bmod N$ and $C_e^a \cdot C^b \cdot h^{-z} = g \bmod N\}$.

(4.1) P computes Damgård-Fujisaki commitments for values a, b, z. That is, P chooses $r_a, r_b, r_z \overset{R}{\leftarrow} \mathbb{Z}_{2^{\gamma+\lambda}}$ and computes the elements $C_a = g^a h^{r_a}$, $C_b = g^b h^{r_b}$, $C_z = g^z h^{r_z}$, all of them modulo N.
P chooses $\alpha_a \in [-2^{k \cdot k_e + \kappa}, 2^{k \cdot k_e + \kappa}]$, $\alpha_b, \alpha_e \in [-2^{k_e+\kappa}, 2^{k_e+\kappa}]$, $\alpha_z \in [-2^{k \cdot k_e + \kappa + \gamma + \lambda}, 2^{k \cdot k_e + \kappa + \gamma + \lambda}]$.
P chooses $\beta_a, \beta_b, \beta_z, \beta_e \in [0, 2^{(\gamma+\lambda+\kappa)}]$.
P computes $Y = C_e^{\alpha_a} C^{\alpha_b} h^{-\alpha_z}$, $F_a = g^{\alpha_a} h^{\beta_a}$, $F_b = g^{\alpha_b} h^{\beta_b}$, $F_z = g^{\alpha_z} h^{\beta_z}$ and $F_e = g^{\alpha_e} h^{\beta_e}$, all of them modulo N.
Finally P sends to V the tuple of values
$$(C_a, C_b, C_z, Y, F_a, F_b, F_z, F_e).$$

(4.2) The verifier V chooses a challenge $c \in [0, 2^\kappa]$ at random and sends it to P.

(4.3) P computes and outputs the integer values $x_a = \alpha_a + ca$, $x_b = \alpha_b + cb$, $x_e = \alpha_e + ce$, $x_z = \alpha_z + cz$, $v_a = \beta_a + cr_a$, $v_b = \beta_b + cr_b$, $v_z = \beta_z + cr_z$ and $v_e = \beta_e + cr$.

The output of the interaction is 1 (accepted) if and only if all the equations below are satisfied, modulo N:

$$\begin{array}{ll} (1) & C_e^{x_a} \cdot C^{x_b} \cdot h^{-x_z} = Y \cdot g^c \\ (2) & g^{x_a} \cdot h^{v_a} = F_a \cdot (C_a)^c \\ (3) & g^{x_b} \cdot h^{v_b} = F_b \cdot (C_b)^c \\ (4) & g^{x_z} \cdot h^{v_z} = F_z \cdot (C_z)^c \\ (5) & g^{x_e} \cdot h^{v_e} = F_e \cdot (C_e)^c \end{array}$$

5.2 Security of the New Protocol

Theorem 1. *Under the Strong QR-RSA Assumption, the previous protocol is a zero-knowledge proof of knowledge of integer values (e, r) such that $C_e = g^e h^r \bmod N$ and such that $e \notin \mathcal{L}$.*

Proof. Completeness of the non-membership proof is straightforward.
In order to prove the zero-knowledge property of the protocol, let us show how a simulator S would proceed, given as inputs a common reference string $\mathsf{crs} = (N, g, h, \gamma, k_e, \kappa, \mathcal{L})$ and a value $C_e = g^e h^r \bmod N$:

1. If $\mathcal{L} = \{e_1, \ldots, e_k\}$, compute $C = g^{e_1 e_2 \ldots e_k} \bmod N$.

2. Now S simulates the 3-moves interaction of step 4 of the real protocol, as follows.
   (2.1) Choose at random $\tilde{x}_a \in [-2^{k \cdot k_e + \kappa + 1}, 2^{k \cdot k_e + \kappa + 1}]$, $\tilde{x}_b, \tilde{x}_e \in [-2^{k_e + \kappa + 1}, 2^{k_e + \kappa + 1}]$, $\tilde{x}_z \in [-2^{k \cdot k_e + \kappa + \gamma + \lambda + 1}, 2^{k \cdot k_e + \kappa + \gamma + \lambda + 1}]$, $\tilde{v}_a, \tilde{v}_b, \tilde{v}_z, \tilde{v}_e \in [0, 2^{(\gamma + \lambda + \kappa + 1)}]$, and finally $\tilde{C}_a, \tilde{C}_b, \tilde{C}_z \in QR_N$.
   (2.2) Choose at random $\tilde{c} \in [0, 2^{\kappa}]$.
   (2.3) Compute $\tilde{Y}, \tilde{F}_a, \tilde{F}_b, \tilde{F}_z, \tilde{F}_e$ in QR_N such that they, along with C_e, C and the values chosen in the two previous steps, satisfy equations (1),. . .,(5) at the end of Section 5.1.

   Define the simulated transcript as
   $tr_{sim} = (\ (\tilde{C}_a, \tilde{C}_b, \tilde{C}_z, \tilde{Y}, \tilde{F}_a, \tilde{F}_b, \tilde{F}_z, \tilde{F}_e),\ \tilde{c},\ (\tilde{x}_a, \tilde{x}_b, \tilde{x}_e, \tilde{x}_z, \tilde{v}_a, \tilde{v}_b, \tilde{v}_z, \tilde{v}_e)\ )$.

It is easy to see that the distribution of tr_{sim} is statistically indistinguishable from the distribution of a transcript $tr \leftarrow \langle P(\mathsf{crs}, e, r), V(\mathsf{crs}, C_e)\rangle$ obtained in a real execution of the protocol, so the zero-knowledge property holds statistically.

Finally, to demonstrate soundness, we use the same technique as in [14]. Let us assume the existence of some prover P^* such that $\varepsilon = \Pr[\langle P^*(\mathsf{crs}, e, r), V(\mathsf{crs}, C_e)\rangle = 1]$, for a honestly generated common reference string $\mathsf{crs} = (N, g, h, \gamma, k_e, \kappa, \mathcal{L})$. Let us construct an extractor E with inputs (crs, C_e) and oracle access to P^* which is able to obtain e, r such that $e \notin \mathcal{L}$ and $C_e = g^e h^r \bmod N$.

The idea for designing E is to use standard rewinding techniques in zero-knowledge interactive arguments: E receives as inputs (crs, C_e) and runs a first execution of the protocol with P^*, which produces a successful transcript tr with probability ε. After that, E rewinds this execution with P^* back until step 4.2, where E (who acts as the verifier V) chooses a challenge $c' \neq c$ different from the challenge c chosen in the initial execution. Again, this second execution produces a successful transcript tr' with probability ε. Note that the elements in the transcripts that correspond to step 4.1 (from C_a to F_e) are equal in the two transcripts. The rest of elements in the transcripts are different, and we denote them as $(c, x_a, x_b, x_e, x_z, v_a, v_b, v_z, v_e)$ for tr and $(c', x'_a, x'_b, x'_e, x'_z, v'_a, v'_b, v'_z, v'_e)$ for tr'.

Summing up, with probability essentially ε^2, the extractor E obtains two transcripts tr, tr' whose elements satisfy the 5 equalities listed at the end of Section 5.1. If the two transcripts satisfy the event Bad, which will be defined below, then E repeats the whole process described up to now, always for the same elements in the first step (4.1), but for new pairs (c, c') of challenges, satisfying $c \neq c'$. This is done until event Bad does not occur.

For instance, regarding equality (2), we have

$$g^{x_a} \cdot h^{v_a} = F_a \cdot (C_a)^c \bmod N \quad \text{and} \quad g^{x'_a} \cdot h^{v'_a} = F_a \cdot (C_a)^{c'} \bmod N.$$

Dividing these two equations, we get $g^{x_a - x'_a} \cdot h^{v_a - v'_a} = (C_a)^{c - c'} \bmod N$, which can be rewritten as

$$g^{(x_a - x'_a) + \alpha(v_a - v'_a)} = (C_a)^{c - c'} \bmod N.$$

By Lemma 1, we have that under the Strong QR-RSA Assumption, it must be the case that $c - c'$ divides $(x_a - x'_a) + \alpha(v_a - v'_a)$. We can define the following events: $\mathsf{E}_1^{(1)}$ is the event where $c - c'$ divides both $x_a - x'_a$ and $v_a - v'_a$, and $\mathsf{Bad}^{(1)} = \neg \mathsf{E}_1^{(1)}$.

Doing the analogous argument with equalities (3), (4) and (5), we get

$$g^{(x_b - x'_b) + \alpha(v_b - v'_b)} = (C_b)^{c - c'} \bmod N,$$

$$g^{(x_z - x'_z) + \alpha(v_z - v'_z)} = (C_z)^{c - c'} \bmod N,$$

$$g^{(x_e - x'_e) + \alpha(v_e - v'_e)} = (C_e)^{c - c'} \bmod N.$$

Applying Lemma 1, we have that $c - c'$ divides the three integer values: $(x_b - x'_b) + \alpha(v_b - v'_b)$, $(x_z - x'_z) + \alpha(v_z - v'_z)$ and $(x_e - x'_e) + \alpha(v_e - v'_e)$. We define the events:

$\mathsf{E}_1^{(2)}$ is the event where $c - c'$ divides both $x_b - x'_b$ and $v_b - v'_b$, $\mathsf{Bad}^{(2)} = \neg \mathsf{E}_1^{(2)}$.

$\mathsf{E}_1^{(3)}$ is the event where $c - c'$ divides both $x_z - x'_z$ and $v_z - v'_z$, $\mathsf{Bad}^{(3)} = \neg \mathsf{E}_1^{(3)}$.

$\mathsf{E}_1^{(4)}$ is the event where $c - c'$ divides both $x_e - x'_e$ and $v_e - v'_e$, $\mathsf{Bad}^{(4)} = \neg \mathsf{E}_1^{(4)}$.

Finally, let us define the event $\mathsf{Bad} = \mathsf{Bad}^{(1)} \vee \mathsf{Bad}^{(2)} \vee \mathsf{Bad}^{(3)} \vee \mathsf{Bad}^{(4)}$.

Let us estimate the probability of event $\mathsf{Bad}^{(j)}$, for $j \in \{1,2,3,4\}$. We consider $\mathsf{Bad}^{(1)}$, the conclusion is the same for the other three cases. When $\mathsf{Bad}^{(1)}$ occurs, we know that $c - c'$ divides $(x_a - x'_a) + \alpha(v_a - v'_a)$ but $c - c'$ does not divide both $x_a - x'_a$ and $v_a - v'_a$. Let $\tilde{q} \geq 2$ be a prime number which divides $c - c'$, and let $n \geq 1$ be the maximum integer number such that $\tilde{q}^n$ divides $c - c'$ but $\tilde{q}^n$ does not divide $x_a - x'_a$; note that we also know that $\tilde{q}^n$ does not divide $v_a - v'_a$ (otherwise, $\tilde{q}^n$ would divide $x_a - x'_a$) and so $v_a - v'_a \neq 0 \bmod \tilde{q}^n$.

Let us write the exponent $\alpha \in \mathbb{Z}_{2^{2\gamma+\lambda}}$ such that $h = g^\alpha \bmod N$ as $\alpha = y + z \cdot \mathrm{order}(g) = y + zq$. Since $c, c' \in [0, 2^\kappa]$ and $2^\kappa \leq q$, we have that $\tilde{q}^n \leq |c - c'| \leq 2^\kappa \leq q$, and so $q \neq 0 \bmod \tilde{q}^n$.

Since $\tilde{q}^n$ divides $c - c'$ and $c - c'$ divides $(x_a - x'_a) + \alpha(v_a - v'_a)$, we can write

$$0 = (x_a - x'_a) + \alpha(v_a - v'_a) = (x_a - x'_a) + y(v_a - v'_a) + zq(v_a - v'_a) \bmod \tilde{q}^n.$$

Since both q and $v_a - v'_a$ are not 0 modulo $\tilde{q}^n$, the number of solutions in $\mathbb{Z}_{\tilde{q}^n}$ of this last equality, for variable z, is at most $\gcd\{\tilde{q}^n, q(v_a - v'_a)\} \leq \tilde{q}^{n-1}$. Since $\alpha = y + zq$ was taken uniformly in $\mathbb{Z}_{2^{2\gamma+\lambda}}$, we have that $z \bmod \tilde{q}^n$ is uniformly distributed in $\{0, 1, \ldots, \tilde{q}^n - 1\}$, and therefore the probability that the equation above is satisfied is bounded by $\frac{\tilde{q}^{n-1}}{\tilde{q}^n} = \frac{1}{\tilde{q}} \leq \frac{1}{2}$.

Summing up, we have $\Pr[\mathsf{Bad}^{(j)}] \leq \frac{1}{2}$ for each $j \in \{1,2,3,4\}$, and therefore we obtain the bound $\Pr[\mathsf{Bad}] \leq \frac{15}{16}$, meaning that the probability that Bad occurs is far from being overwhelming. This means that, after a polynomial number or repetitions, the extractor E gets a pair of valid transcripts such that event Bad does not occur.

In such a case, since $\mathsf{E}_1^{(j)}$ occurs for all $j \in \{1,2,3,4\}$, we have that $c - c'$ divides all the integers $x_a - x'_a$, $v_a - v'_a$,. . .,$x_e - x'_e$ and $v_e - v'_e$. In particular, we obtain integers $\hat{a} := \frac{x_a - x'_a}{c - c'}$, $\hat{b} := \frac{x_b - x'_b}{c - c'}$, $\hat{z} := \frac{x_z - x'_z}{c - c'}$, $e := \frac{x_e - x'_e}{c - c'}$, $r := \frac{v_e - v'_e}{c - c'}$. Note that the two last values (e, r), when inserted into the equality

$$g^{(x_e - x'_e) + \alpha(v_e - v'_e)} = (C_e)^{c - c'} \bmod N,$$

lead to $(g^e \cdot h^r)^{c - c'} = (C_e)^{c - c'} \bmod N$. Remember that the order of g, h, C_e is the prime q. If $c - c'$ was not coprime to q, then E could factorize N. Therefore, with overwhelming probability $c - c'$ is coprime to q and so we get the equality $g^e \cdot h^r = C_e \bmod N$.

Now we focus on the two instances of equality (1) and we divide them, obtaining

$$(C_e)^{x_a - x'_a} \cdot C^{x_b - x'_b} \cdot h^{-x_z + x'_z} = g^{c - c'} \bmod N,$$

which can be rewritten as

$$\left(C_e^{\hat{a}} \cdot C^{\hat{b}} \cdot h^{-\hat{z}}\right)^{c - c'} = g^{c - c'} \bmod N.$$

As argued above, with overwhelming probability we have that $c - c'$ is coprime to q, and so we get the equality $C_e^{\hat{a}} \cdot C^{\hat{b}} \cdot h^{-\hat{z}} = g \bmod N$ in QR_N. Recall that $C_e = g^e \cdot h^r \bmod N$ and that $C = g^{e_1 e_2 \cdots e_k} \bmod N$. Putting these pieces together, we get

$$g^{\hat{a}e} \cdot h^{\hat{a}r} \cdot g^{\hat{b} \prod\limits_{i=1}^{k} e_i} \cdot h^{-\hat{z}} = g \bmod N.$$

Denoting the integer exponents $s := \hat{a}e + \hat{b}\left(\prod\limits_{i=1}^{k} e_i\right) - 1$ and $t := \hat{a}r - \hat{z}$, the last equality becomes $g^s \cdot h^t = 1 \bmod N$, or in other words $g^s = h^{-t} \bmod N$. If $t \neq 0$, then we can apply Lemma 1 to conclude that, under the Strong RSA Assumption, $\ell := \frac{s}{-t} \in \mathbb{Z}$ is an integer value. The case $\ell = \pm 1$ would mean $h = g$ or $h = g^{-1} \bmod N$, which happens with negligible probability. Otherwise, when $|\ell| > 1$, we get $h = h^{\frac{-t\ell}{s}} = g^\ell \bmod N$, so E could be used to solve the Strong RSA problem for $\omega = h$, which would contradict the Strong RSA Assumption.

The only remaining possibility is $t = 0$, which leads to $g^s = 1 \bmod N$ and so $s = 0 \bmod q$. Recovering the value of s, we have

$$\hat{a}e + \hat{b}\left(\prod_{i=1}^{k} e_i\right) = 1 \bmod q. \tag{1}$$

If it was the case $e \in \mathcal{L} = \{e_1, \ldots, e_k\}$, namely $e = e_{i^*}$ for some $i \in \{1, \ldots, k\}$, then equality (1) would become

$$1 = e_{i^*} \cdot \left(\hat{a} + \hat{b}\left(\prod_{1 \leq i \leq k, i \neq i^*} e_i\right)\right) \bmod q \tag{2}$$

Since all the elements in $\mathcal{L}$ are prime numbers, we have $e_{i^*} > 1$ and so the right term of equality (2) would be strictly larger than 1, which would allow E to obtain q, and thus the factorization of N, which is a contradiction with the Strong QR-RSA Assumption.

The conclusion of this case analysis is that the only valid possibility, under the Strong QR-RSA Assumption, is $t = 0$ and $e \notin \mathcal{L}$. Therefore, as desired, the extractor E obtains, with probability ε^2, two integer values e, r such that $C_e = g^e h^r \bmod N$ and such that $e \notin \mathcal{L}$. □

Although we have proved the soundness property of the new zero-knowledge protocol with respect to the Strong (QR) RSA Assumption, it is possible to use the recent techniques developed in [12] to obtain a proof of soundness with respect to the (weaker and more standard) RSA Assumption.

6 Applications, Efficiency Analysis and Comparisons

In this section we discuss the applicability of the new protocol to add secure revocation in scenarios where anonymous authentication is implemented by using RSA cryptographic techniques. We also compare the efficiency of the new protocol with that of the protocol proposed in [20].

6.1 Where and How Can the Protocol be Used?

The new zero-knowledge protocol for non-membership can be combined with many existing anonymous authentication systems that use RSA-based cryptographic techniques, in order to provide revocation, as long as the secret authentication information of each user U_i contains a prime number e_i different from those of the other users. Therefore, the new protocol can be applied exactly in the same situations as the zero-knowledge protocol of Li *et al.* [20]. Examples of such authentication systems are the group signature scheme of Ateniese *et al.* [1], the attribute-based signature scheme of Herranz [17], the anonymous credential system of Camenisch and Lysyanskaya [7, 8] (which is at the core of the Identity Mixer product, Idemix, by IBM [18]) or the dynamic anonymous attestation system of Brickell *et al.* [5].

Both our new protocol and the protocol in [20] can be implemented in two different ways:

- *Option 1: $\mathcal{L}$ is kept secret.* The motivation to choose this option could be the fact that the size of the public parameters of the system stay constant, independent of the number of revoked users (the size of $\mathcal{L}$). In this scenario, the system administrator manages the list $\mathcal{L}$ and publishes only the associated accumulator, $C = g^{\prod\limits_{e_i \in \mathcal{L}} e_i} \bmod N$, which has constant size. Another possible advantage is the fact that the numbers e_i associated to the revoked users remain hidden, which seems to provide anonymity to users even when they are revoked. The drawback of this approach is that, every time the list $\mathcal{L}$ changes (because new users are revoked), the system administrator is the only one who can update the value of C and the non-membership witnesses of all the non-revoked users; the updated witnesses must be sent to those users in a secret way. Therefore, this scenario has a strong connectivity requirement.

- *Option 2: $\mathcal{L}$ is made public.* The only task of the system administrator is to update the (public) list $\mathcal{L}$ and the corresponding accumulator C. Now the users can update the non-membership witnesses by themselves, each time the list $\mathcal{L}$ changes, without any (private) interaction with the system administrator. The fact of publishing $\mathcal{L}$ results in a larger set of public parameters. However, in some situations, like authentication systems for small companies, the number of revoked users is expected to be small. In applications (like ePassports or eID cards) with a larger expected number of revocations, the total number of (non-revoked) users is also huge, so still the drawback of publishing $\mathcal{L}$ could be preferable to the drawback of requiring interaction between the system administrator(s) and users each time $\mathcal{L}$ changes. Note that this interaction (connectivity) issue is considered to be the main drawback of accumulator-based revocation solutions for Idemix, in the comparative analysis provided in [19].
  Finally, how do the anonymity properties of the authentication system behave when $\mathcal{L}$ is made public? In all the authentication systems listed above, the anonymity property is so strong that even with the knowledge of e_i it is impossible to trace or link the executions of the authentication protocol that were executed using e_i. Therefore, publishing $\mathcal{L}$ does not affect the anonymity of the system: non-revoked users keep the same anonymity as before, and revoked users know that their authentication actions before revocation remain anonymous even if future authentication actions will not succeed.

6.2 Efficiency Comparison with [20]

The zero-knowledge non-membership argument proposed by Li *et al.* in [20] to prove that a committed element e is not accumulated in the value C consists of 6 steps: the first one is a step where the prover computes and sends 6 elements in QR_N, and the other five steps are five interactive (3 moves) protocols that can be run in parallel. Since the new protocol proposed in this paper is also a 3 moves interactive protocol, we can transform the two solutions into non-interactive solutions by applying the well-known Fiat-Shamir heuristic [15], which replaces the challenge value c chosen by the verifier with the use of a hash function that the prover applies to all the elements computed in the first move.

For instance, in the case of the protocol proposed in this work, the non-interactive proof would have the form

$$\sigma = (C_a, C_b, C_z, c, x_a, x_b, x_e, x_z, v_a, v_b, v_z, v_e).$$

Recall that we are assuming that the size of e and each $e_i \in \mathcal{L} = \{e_1, \ldots, e_k\}$ is at most k_e bits. This proof contains 3 elements in QR_N, which are λ-bits long each; one element c which is κ-bits long; and 8 integer values with different sizes, because the sizes of the underlying secret integers $(a, b, e, r, z = ar, \ldots)$ are different. Namely, $|a| \leq \left(2^{k_e}\right)^k$, whereas $e, |b| \leq 2^{k_e}$, and so on. The size of v_a, v_b, v_z, v_e is the same. Taking into account these bounds, we have that the length in bits of the non-interactive proof produced by our protocol would be

$$4\lambda + \kappa + k(k_e + \kappa) + 2\,(k_e + \kappa) + k(k_e + \gamma + \lambda + \kappa) + 4\,(\gamma + \lambda + \kappa)\,.$$

In contrast, the non-interactive proof produced by the protocol in [20] contains 10 elements in QR_N (the 6 elements in the initial step, plus 4 elements for a range proof [3]); one element c (the output of the hash function) which is κ-bits long; and 21 integers with different sizes (all of them independent of the cardinality k of the list $\mathcal{L}$ of revoked users). Taking into account the bounds for those integers, we have that the length in bits of the non-interactive proof produced by the protocol in [20] would be

$$10\lambda + \kappa + 6\,(k_e + \kappa) + 2(k_e + \lambda + \kappa) + 12\,(\gamma + \lambda + \kappa) + (\lambda + \kappa).$$

Therefore, we have that the non-interactive zero-knowledge proofs of non-membership produced by our protocol and by the protocol in [20] contain the following amount of bits:

$$\begin{array}{rrrrrl} \ell_1 = & (8+k)\lambda\, + & (4+k)\gamma\, + & (2+2k)k_e\, + & (7+2k)\kappa & \text{(ours)} \\ \ell_2 = & 25\lambda\, + & 12\gamma\, + & 8k_e\, + & 22\kappa & \text{([20])} \end{array}$$

A similar result is obtained if we compare the running times (dominated by the modular exponentiations) required for either producing or verifying such a proof, because the cost of a modular exponentiation depends linearly on the number of bits of the integer exponents.

Which protocol is preferable? The complexity of the protocol in [20] is independent of the cardinality k of the list $\mathcal{L}$ of revoked users, as opposed to what happens with the complexity of our protocol. However, the constants that appear in the complexity measures are smaller in the case of our new protocol. This means that the new protocol is preferable in situations where the list of revoked users is expected to be small, and the protocol in [20] is preferable in situations with potentially many revoked users.

We illustrate this effect with the particular case of the product Idemix [18]; the cryptographic primitive underlying Idemix is the anonymous credential system in [9], which is based on the group signature / identification schemes in [1]. In the setting proposed therein, the RSA modulus N is λ bits long and the prime numbers e_i that are part of the secret key of the users are k_e bits long, where $k_e > \lambda + 2\kappa$. Taking for simplicity $\gamma = \lambda$ and $k_e = \lambda + 2\kappa$, we get $\ell_1 = (14+4k)\lambda + (11+6k)\kappa$ and $\ell_2 = 45\lambda + 38\kappa$.

This means that the proofs produced by our new protocol are shorter than those produced by [20] only when the number of revoked users is $k \leq 7$. Although this could seem a very particular situation, we remark than in some settings, like anonymous access control to (physical or digital) resources with few users, the number of revoked (in other words, misbehaving) users is expected to be small. In such a case, for instance if the number of revoked users is $k = 2$, our new protocol would produce proofs which are twice shorter than those produced by the protocol in [20].

Furthermore, since the two protocols work on the same mathematical setting and share the same public parameters (the RSA modulus, the generators elements $g, h \in QR(N)$, etc.), one possibility is that the anonymous authentication system (e.g. Idemix) admits the two non-membership protocols, so that our new protocol can be used when the number of revoked users is small and the protocol in [20] is used once the number of revoked users exceeds some threshold, for instance $k = 7$.

7 Conclusions

A lot of RSA-based anonymous authentication systems have been proposed, including group and attribute-based signatures, anonymous credential systems, or direct anonymous attestation. A notable example is the Identity Mixer (Idemix) product [18] by IBM, which can potentially be integrated into electronic ID cards and electronic passports. Revocation of users due to bad behavior or loss of secret keys is a necessary but delicate requirement for such systems. There are different strategies to provide this property of revocation, see for instance [19] for the specific case of Idemix: therein, the conclusion is that the best strategy may be the use of accumulator-based (non-)membership zero-knowledge arguments, specifically those in [9, 20]. However, the efficiency of these two last protocols seems far from being optimal, because they are quite complex and require a lot of operations.

Peng and Bao [21] already tried to design a more efficient zero-knowledge non-membership argument for such a specific setting. We have proved in this paper that their protocol is not secure, and we have designed a different protocol which has provable security. Finally, we have compared the efficiency of our new protocol with that of the protocol in [20], in particular when used in the setting of the product Idemix, to conclude that the new protocol is more efficient in situations where the number of revoked users is small.

As a future work, we would like to investigate other possible ways to improve the efficiency of these two non-membership zero-knowledge protocols (the one in [20] and the one proposed here); a promising approach may be the use of the techniques recently introduced in [12]. However, these techniques seem to require a strong interaction between the prover and the verifier, which would limit the use of the resulting revocation mechanisms to interactive authentication systems.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. *Proc. of Crypto'00*, LNCS **1880**, Springer-Verlag, pp. 255–270 (2000)
2. S. Bayer and J. Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. *Proc. of Eurocrypt'13*, LNCS **7881**, Springer-Verlag, pp. 646–663 (2013)
3. F. Boudot. Efficient proofs that a committed number lies in an interval. *Proc. of Eurocrypt'00*, LNCS **1807**, Springer-Verlag, pp. 431–444 (2000)
4. S. Brands, L. Demuynck, and B. De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. *Proc. of ACISP'07*, LNCS **4586**, Springer-Verlag, pp. 400–415 (2007)
5. E.F. Brickell, J. Camenisch and L. Chen. Direct anonymous attestation. *Proc. of CCS'04*, ACM Press, pp. 132–145 (2004)
6. J. Camenisch, M. Kohlweiss and C, Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. *Proc. of PKC'09*, LNCS **5443**, Springer-Verlag, pp. 481–500 (2009)
7. J. Camenisch and A. Lysynskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *Proc. of Eurocrypt'01*, LNCS **2045**, Springer-Verlag, pp. 93–118 (2001)
8. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. *Proc. of SCN'02*, LNCS **2576**, Springer-Verlag, pp. 268–289 (2003)
9. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. *Proc. of Crypto'02*, LNCS **2442**, Springer-Verlag, pp. 61–76 (2002)
10. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. *Proc. of Crypto'97*, LNCS **1294**, Springer-Verlag, pp. 410–424 (1997)
11. D. Catalano and D. Fiore. Vector commitments and their applications. *Proc. of PKC'13*, LNCS **7778**, Springer-Verlag, pp. 55–72 (2013)
12. G. Couteau, T. Peters and D. Pointcheval. Removing the Strong RSA Assumption from arguments over the integers. IACR ePrint Archive, `https://eprint.iacr.org/2016/128` (2016)
13. R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. *ACM Transactions on Information and System Security*, Volume **3** (Issue 3), ACM Press, pp. 161–185 (2000)
14. I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. *Proc. of Asiacrypt'02*, LNCS **2501**, Springer-Verlag, pp. 125–142 (2002)
15. A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. *Proc. of Crypto'86*, LNCS **263**, Springer-Verlag, pp. 186–194 (1986)
16. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. *Proc. of Crypto'97*, LNCS **1294**, Springer-Verlag, pp. 16–30 (1997)
17. J. Herranz. Attribute-based signatures from RSA. *Theoretical Computer Science*, Volume **527**, Elsevier, pp. 73–82 (2014)
18. Identity Mixer, IBM Zurich Research Laboratory. `http://www.zurich.ibm.com/idemix/` (2015)
19. J. Lapon, M. Kohlweiss, B. De Decker and V. Naessens. Analysis of revocation strategies for anonymous Idemix credentials. *Proc. of CMS'11*, LNCS **7025**, Springer-Verlag, pp. 3–17 (2011)
20. J. Li, N. Li and R. Xue. Universal accumulators with efficient nonmembership proofs. *Proc. of ACNS'07*, LNCS **4521**, Springer-Verlag, pp. 253–269 (2007)
21. K.Peng and F. Bao. Improving applicability, efficiency and security of non-membership proof. *Proc. of ISDPE'10*, IEEE Society Press, pp. 39–44 (2010)