Using the RSA or RSA-B accumulator in anonymous credential schemes

Sietse Ringers

September 14, 2023

Abstract

We review the two RSA-based accumulators introduced by Camenisch and Lysyanskaya in [CL02b] in the setting of revocation for anonymous credential schemes, such as Idemix or BBS+. We show that in such a setting, the lower and upper bounds placed on the accumulated values in the paper are unnecessarily strict; they can be removed almost entirely (up to the group order of the credential scheme). This allows the accumulators to be used on elliptic curves of ordinary sizes, such as the ones on which BBS+ is commonly implemented. We also offer some notes and optimizations for implementations of anonymous credential schemes that use these accumulators to enable revocation.

Contents

1 Introduction

The RSA-based accumulator schemes from [CL02b] allow anonymous credentials such as those from Identity Mixer (Idemix) [CL02a; IBM12] or BBS+ [ASM06; CDL16] to be revoked, without compromising the anonymity features of the credential scheme. Camenisch and Lysyanskaya introduce two accumulators in [CL02b] that work in an RSA-like setting, one of which is such that it has a trivial addition operation: only deleting values from the accumulator constitutes work. We follow the lead of [Bal+17] and refer to the two accumulator schemes as the RSA and RSA-B accumulators.

Let $\mathfrak{q}$ be the order of the group of the anonymous credential scheme. In [CL02b], the prime numbers e that are accumulated have to be chosen from a set contained within $[A, B]$ which is such that $B2^{k'+k''+2} < A^2 - 1 < \mathfrak{q}/2$, in which k' is the bit length of challenges chosen in the zero-knowledge proofs and k'' is the statistical zero-knowledge security parameter of the zero-knowledge proofs. In practice one would want $k'' = 128$, and since one generally uses the SHA256 hash function in the Fiat-Shamir heuristic to compute the challenges, whose output is 256 bits, this means that $\mathfrak{q}$ has to be at the very minimum $128 + 256 + 2 + 1 = 387$ bits. However, if one wants the set $[A, B]$ to have a reasonable size, for example so large that one can randomly choose primes from them with a negligible probability for collisions, the birthday paradox combined with the prime counting theorem will push this minimum size for $\mathfrak{q}$ to at least some 500 bits, and probably larger. This would make it impractical to use these accumulators in generally available elliptic curves, which are currently usually between 256 and some 400 bits. That is, using these accumulators in existing BBS+ implementations would be impossible.

In this paper, we show that both the lower bound A and upper bound B can be relaxed to such a degree that this becomes feasible again. In addition, we suggest a number of optimizations that can be made in implementations. The contents of this paper largely follows that of [CL02b], to which we will sometimes refer as just "the paper". At some points we remark in footnotes on minor notational mistakes in the paper, or small differences between it and this paper.

The outline of this paper is as follows.

- In Section 2, we show that the lower bound A does not need to be related to the upper bound B. Instead, it suffices to require that $A > 1$.
- In Section 3, we review the proof of security for the zero-knowledge proof of a witness for an accumulated value, adding extra

explanation here and there and fixing some minor notational errors present in [CL02b].

- In Section 4 we apply the accumulator to anonymous credential schemes, and we show that in such a setting the upper bound B does not need to satisfy $B2^{k'+k''+2} < \mathfrak{q}/2$; instead requiring $B < \mathfrak{q}$ suffices.
- In Section 5, we suggest a number of optimizations that can be made when implementing the accumulator for BBS+ and Idemix.
- In Section 6, we finally consider the differences between the RSA and RSA-B accumulators, ending with the conclusion that RSA-B is superior. (Up to that point in the paper we restrict our attention to the RSA accumulator, to stay closer in our considerations and notation to [CL02b].)

We keep our notation as close as possible to that of [CL02b]. If a is an integer we denote with $a \bmod b$ the remainder of division of a by b, i.e., the unique integer $r < b$ such that $a = \lfloor a/b \rfloor b + r$. The modulus $n = pq$ is a product of two safe primes; that is, writing $p = 2p'+1$ and $q = 2q'+1$, p' and q' are also primes. $\mathbb{Z}_n^* = (\mathbb{Z}/n\mathbb{Z})^*$ is the multiplicative group of integers modulo n, and $QR_n \subset \mathbb{Z}_n^*$ is the subgroup of quadratic residues, i.e., $QR_n = (\mathbb{Z}_n^*)^2 = \{x \in \mathbb{Z}_n^* \mid \exists y \in \mathbb{Z}_n^* \colon x = y^2\}$, whose order is $p'q'$. When dealing with elements of QR_n or $\mathbb{Z}_n$ we often omit writing $\bmod n$ after multiplication, exponentiation, or modular division; this is implied (although when we work with groups of other moduli we will be more careful with this to avoid confusion).

We denote with ν the accumulator. In the RSA accumulator a prime number e may be accumulated into ν by setting $u = \nu$ and then $\nu \mapsto \nu^e$. In the RSA-B accumulator, the number u is instead calculated by $u = \nu^{e^{-1} \bmod p'q'}$, and the value of the accumulator ν stays the same. The number $u \in QR_n$, which in both cases satisfies $\nu = u^e$ by construction, is called the witness for e. The number e can be removed from the accumulator by $\nu \mapsto \nu^{e^{-1} \bmod p'q'}$.

2 The lower bound of accumulated values

Theorem 3 in Section 3.2 of [CL02b] proves security of the RSA accumulator, assuming that the set $X_{A,B} \subset [A, B]$ from which the primes e are chosen is such that $B < A^2$. In this section, we show that this assumption is not necessary: it is possible to prove security without it. This allows us to choose these parameters smaller than they would otherwise need to be, increasing the efficiency of the scheme.

Recalling the definition of security of an accumulator from the paper, we say that an accumulator f, which adds a value e to the accumulator ν by $\nu \mapsto f(\nu, e)$, is secure when the following holds. Let $\mathcal{X}$ be the set of values that may be accumulated, and let $\mathcal{X}'$ be the range of the second parameter of f (so that $\mathcal{X} \subset \mathcal{X}'$). For any ν, no probabilistic polynomial-time algorithm can compute a subset $X = \{x_1, \dots, x_n\} \subset \mathcal{X}$ as well as some $x \in \mathcal{X}'$ and u such that $x \notin X$ and $f(u, x) = f(\nu, X)$ (where with $f(\nu, X)$ we mean $f(f(\cdots(f(\nu, x_1), \dots), x_n)$.)

Note that this definition says that it must be impossible to come up with any element from the larger set $\mathcal{X}'$, which contains $\mathcal{X}$ as a subset, as a "forgery". In the case of the RSA(-B) accumulator, this means any integer x unequal to $\pm 1 \bmod n$ such that $u^x = \nu^X$. However, the legitimately accumulated values $x_i \in X$ are elements of $\mathcal{X}$.

Theorem 1. *Let $X_{A,B} \subset [A, B]$ with $A > 1$ be the set of numbers from which the number e such that $\nu = u^e$ is chosen. Under the strong RSA assumption, the RSA accumulator is a secure dynamic accumulator.*

Proof. We set everything up just as in the first part of the proof of Theorem 3 in the paper.[1] In particular, let u be the number of which we wish to compute a root, and suppose the adversary came up with numbers $u', x' \neq \pm 1 \bmod n$, and $x_1, \dots, x_k \in X_{A,B}$ such that $x' \neq x_i$ for all k and

$$(u')^{x'} = u^x,$$

in which $x = \prod_{i=1}^{k} x_i$. The first step is to derive from this a similar equation in which the exponents are relatively prime.

Set $d = \gcd(x, x')$ and let $y = x/d$ and $y' = x'/d$. Then $(u')^{x'} = u^x$ becomes

$$((u')^{y'})^d = (u^y)^d.$$

If d is even then $(u')^{y'}$ need not be a quadratic residue, but by Lemma 1 we can always write $(u')^{y'} = sr \bmod n$ for some $s^2 = 1 \bmod n$ and

[1] This proof differs from the one in the paper in the following respects. (1) We place no requirements on the subset $X_{A,B} \subset [A, B]$ except for $A > 1$: any number as long as it exceeds 1 may be accumulated, including composites. (2) We are more explicit in handling the square root s of 1 that necessarily shows up when relating elements of $\mathbb{Z}_n^*$ and $QR_n \subset \mathbb{Z}_n^*$ with each other. (3) To ease notational burden, we write y and y' instead of $\tilde{x}$ and $\tilde{x}'$ throughout the proof.

$r \in QR_n$. Then

$$\begin{aligned}
(u^y)^d &= ((u')^{y'})^d \bmod n \\
&= (sr)^d \bmod n \\
&= s^d r^d \bmod n \\
&= r^d \bmod n,
\end{aligned}$$

where the last step follows because the left hand side shows that we are comparing quadratic residues. Now u^y and r are both quadratic residues, and since $d \bmod \#QR_n$ is invertible by Lemma 4, the outer ends of the above equations imply

$$u^y = r \bmod n = s(u')^{y'} \bmod n \quad \text{with} \quad s^2 = 1 \bmod n.$$

By Lemma 3 the number s must be ± 1. If y' is even, then $s = 1$ since then both u^y and $(u')^{y'}$ are elements of QR_n, while $-1 \notin QR_n$ (as -1 is of order 2). If y' is odd, then we may move s within the brackets to obtain $u^y = (su')^{y'}$ for $s = \pm 1$. Setting $v = su'$, in all of these cases we have

$$u^y = v^{y'} \bmod n \quad \text{with} \quad \gcd(y, y') = 1,$$

since $d = \gcd(x, x') = \gcd(yd, y'd)$.

The rest of the proof is the same as in the paper. Using the extended Euclidian algorithm for gcd we can compute integers a, b such that $ay + by' = 1$ holds. Set $z = v^a u^b$. Then[2]

$$z^{y'} = (z^{yy'})^{1/y} = \left((v^{y'})^{ay}(u^y)^{by'}\right)^{1/y} = \left((u^y)^{ay+by'}\right)^{1/y} = u;$$

note that $1/y \bmod \#QR_n$ exists again by Lemma 4. Thus the tuple (z, y') breaks the strong RSA assumption. $\square$

The claim that $d = \gcd(x, x')$ implies $d = 1$ or $d = x_j$ found in the middle of the proof in the paper, which requires $A^2 > B$ for its proof, is thus not necessary to use in the proof of Theorem 3. The paper uses this to infer that d is invertible modulo $\#QR_n$, but Lemma 4 shows that under the Strong RSA assumption any number computed by a polynomial-time algorithm is with overwhelming probability going to be invertible modulo $\#QR_n$, regardless of their size or how they are constructed.

Going further, for this proof to work it is not even necessary to require that the set $X_{A,B}$ of values e that may be accumulated must consist only of primes, since this proof nowhere uses that the numbers x_i are prime. Instead, any numbers $x_i > 1$ may be used.

[2]The paper writes $\tilde{u}$ instead of u' here, and it misses the second and last equality signs.

The corresponding proof in the paper seems to use the fact that the numbers e are prime only in the Claim inside the proof, which as noted is not necessary to use. This is the only place in the paper where the requirement that the numbers e are prime numbers is used; all other security results about the RSA(-B) accumulators ultimately reduce to this one. Therefore it seems unnecessary to require the accumulated numbers e to be prime.

In the remainder of this document, therefore, we drop these requirements; instead we only require that $A > 1$. What B needs to satisfy is discussed in Section 4.

3 The proof of knowledge

When using accumulators one generally wishes to prove that something is not revoked. In this paper we assume that we are dealing with an anonymous credential scheme, such as the Identity Mixer (Idemix) [CL02a; IBM12] or BBS+ [ASM06; CDL16]. In such cases, for the system to work we have to tie the witness (u, e) to the credential that needs to be revocable. This can be done by including the number e as one of the attributes signed by the issuer, and during verification letting the user prove equality of the signed attribute and the number e such that $u^e = \nu$.

Generally, such an issuer signature takes the form of a signature over a commitment to e and the credential's attributes, in the group $\mathfrak{G}$ in which the credential scheme lives. The precise form of this commitment will depend on the details of the credential scheme. The particular identities being zero-knowledge proved in what follows may have to be correspondingly adjusted, according to analysis of the specific form of the commitment to e and the attributes, the signature over that, and the group structure of $\mathfrak{G}$ (in particular, its order $\mathfrak{q}$). Indeed, the paper does this too in its example application of the accumulator, in section 4.2. We will come back to this in Sections 4 and 5.

For now we ignore all details of the credential scheme being used by assuming, like the paper, that the commitment to e in $\mathfrak{G}$ is of the form $\mathfrak{C}_e = \mathfrak{g}^e \mathfrak{h}^r$, where $\mathfrak{g}, \mathfrak{h} \in \mathfrak{G}$. We denote with $\mathfrak{q}$ the order of $\mathfrak{G}$. Note that any commitment in $\mathfrak{G}$ can then only commit to integers reduced modulo $\mathfrak{q}$.

To prove knowledge of u and e such that $\nu = u^e$ and the number $e \bmod \mathfrak{q}$ is committed to by $\mathfrak{C}_e$, the prover forms a commitment C_u to u and proves that this commitment corresponds to an e-th root of the value ν. This is carried out as follows.

- Denote with k'' the statistical zero-knowledge security parameter.

The prover chooses $r_1, r_2, r_3 \in \mathbb{Z}_{\lfloor n/4 \rfloor}$ (i.e., having as upper bound the largest number known by everyone that is below the order of QR_n), computes $C_e = g^e h^{r_1}$, $C_u = uh^{r_2}$, $C_r = g^{r_2} h^{r_3}$, and sends $\mathfrak{C}_e$, C_e, C_u and C_r to the verifier.

- The prover and verifier carry out the following proof of knowledge:[3, 4]

$$\begin{aligned}
\mathrm{PK}\Bigg\{ &(\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \varphi, \psi, \eta, \sigma, \xi) : \\
&\mathfrak{C}_e = \mathfrak{g}^{\alpha} \mathfrak{h}^{\varphi} \ \wedge \ \mathfrak{g} = \left(\frac{\mathfrak{C}_e}{\mathfrak{g}}\right)^{\gamma} \mathfrak{h}^{\psi} \ \wedge \ \mathfrak{g} = (\mathfrak{g}\mathfrak{C}_e)^{\sigma} \mathfrak{h}^{\xi} \ \wedge \\
&\qquad C_r = g^{\epsilon} h^{\zeta} \ \wedge \ C_e = g^{\alpha} h^{\eta} \ \wedge \\
&\qquad \nu = C_u^{\alpha} \left(\frac{1}{h}\right)^{\beta} \ \wedge \ 1 = C_r^{\alpha} \left(\frac{1}{g}\right)^{\delta} \left(\frac{1}{h}\right)^{\beta} \Bigg\}
\end{aligned} \tag{1}$$

In these expressions, for honest users the following would hold:

- $\alpha = e$
- $\beta = er_2$
- $\varphi = r$
- $\eta = r_1$
- $\zeta = r_2$
- $\epsilon = r_3$

We will however keep the paper's notation and use the greek letters.

The reason for the fractions in these expressions is to force various relations between these numbers that are required to convince the verifier, as will become clear in the proof.

The identities above require us to prove knowledge of exponents involving the group $\mathfrak{G} \ni \mathfrak{g}, \mathfrak{h}$ which is of order $\mathfrak{q}$, as well as the group QR_n which is of the unknown order $p'q'$. Proving knowledge in the latter relies on not reducing the response modulo the unknown $p'q'$. On the other hand, proving knowledge of an exponent involving $\mathfrak{G}$ necessarily requires the prover as well as the verifier to perform reductions modulo $\mathfrak{q}$ of all exponents.

[3] In the second line of this proof in the paper, the g and h factors are erroneously switched. We can tell that this is an error and not done on purpose because α plays the role of e, and C_e is defined as $C_e = g^e h^{r_1}$, which is inconsistent with $C_e = h^{\alpha} g^{\eta}$ in the second line of the PK in the paper. We have swapped g and h to their expected order, and swapped some of the greek indices as well in such a way that in the proof, the greek letters have to change as little as possible.

[4] Contrary to the corresponding proof in the paper our proof gives no guarantees on the size of the number α, because we do not need it in our use of this proof later in the paper, in Section 4.

In particular, the number α occurs in each of the lines in the equalities above, as exponents involving $\mathfrak{G} \ni \mathfrak{g}, \mathfrak{h}$ as well as QR_n. The zero-knowledge proof must be able to convince the verifier that the prover uses one and the same number α in each of the expression where α occurs, up to reduction modulo $\mathfrak{q}$ in expressions that involve $\mathfrak{G}$. As one normally does, we implement this by having the prover send a single response s_α that the verifier uses for all identities involving α. This is an integer, i.e., not reduced modulo $\mathfrak{q}$ or anything else. However, as noted above, if the identity involves $\mathfrak{G}$ then in computations a reduction modulo $\mathfrak{q}$ of α and s_α is implied.

Theorem 2. *Under the strong RSA assumption this is a proof of knowledge of two integers $e \neq 0, \pm 1$ and u such that $\nu = u^e \bmod n$ and $\mathfrak{C}_e$ is a commitment to $e \bmod \mathfrak{q}$.*

Proof. Showing that the protocol is statistical zero-knowledge is standard. Also, it is easy to see that $\mathfrak{C}_e$, C_e, C_u and C_r are statistically independent from u and e.

It remains to show that if the verifier accepts, then numbers $e \neq 0, \pm 1$ and u such that (1) $\mathfrak{C}_e$ commits to $e \bmod \mathfrak{q}$ and (2) $u^e = \nu$ can be extracted from the prover. We do this with the standard rewinding technique, i.e., presenting the prover with different challenges c and c', and using its output to construct such numbers. We proceed as follows.

- The first line of the proven equalities in the proof of knowledge (1) involving $\mathfrak{C}_e$ allows us to extract the modular number $e \bmod \mathfrak{q}$ that $\mathfrak{C}_e$ commits to, and conclude that it is unequal to $\pm 1 \bmod \mathfrak{q}$.
- The second and third lines allow us to extract integers u, e such that (1) $\nu = u^e$ and (2) the reduction modulo $\mathfrak{q}$ of e equals the number extracted in the previous step.

Therefore we may conclude that $e \neq \pm 1$ as integers, because otherwise $e \neq \pm 1 \bmod \mathfrak{q}$. Additionally, since $\nu \neq 1$ we have $e \neq 0$.

In the remainder of the proof, we drop the variables u and e and work exclusively with what the adversary gave us. First we set up some notation.

- Denote with s_α and s'_α the responses for α that the prover emits when presented with c and c', respectively. Similarly for all other greek letters.
- Set $\Delta\alpha = s_\alpha - s'_\alpha$, and similarly for all other greek letters. Additionally, set $\Delta c = c' - c$.
- Set $\widetilde{\alpha} = \Delta\alpha\Delta c^{-1} \bmod \mathfrak{q}$, and similarly for all other greek letters, for the relations involving $\mathfrak{G}$.

Then after completing the proof, we have

$$\mathfrak{C}_e^{\Delta c} = \mathfrak{g}^{\Delta\alpha}\mathfrak{h}^{\Delta\varphi} \qquad \mathfrak{g}^{\Delta c} = \left(\frac{\mathfrak{C}_e}{\mathfrak{g}}\right)^{\Delta\gamma}\mathfrak{h}^{\Delta\psi} \quad \mathfrak{g}^{\Delta c} = (\mathfrak{g}\mathfrak{C}_e)^{\Delta\sigma}\mathfrak{h}^{\Delta\xi} \tag{2}$$

$$C_r^{\Delta c} = g^{\Delta\epsilon}h^{\Delta\zeta} \qquad C_e^{\Delta c} = g^{\Delta\alpha}h^{\Delta\eta} \tag{3}$$

$$\nu^{\Delta c} = C_u^{\Delta\alpha}\left(\frac{1}{h}\right)^{\Delta\beta} \qquad 1 = C_r^{\Delta\alpha}\left(\frac{1}{g}\right)^{\Delta\delta}\left(\frac{1}{h}\right)^{\Delta\beta} \tag{4}$$

We first show that $\mathfrak{C}_e$ commits to a number different from $\pm 1 \bmod \mathfrak{q}$, using (2). The left equation yields $\mathfrak{C}_e = \mathfrak{g}^{\widetilde{\alpha}}\mathfrak{h}^{\widetilde{\varphi}}$ and the middle yields $\mathfrak{g} = (\mathfrak{C}_e/\mathfrak{g})^{\widetilde{\gamma}}\,\mathfrak{h}^{\widetilde{\psi}}$, and substituting the one in the other results in[5]

$$\mathfrak{g} = \left(\frac{\mathfrak{C}_e}{\mathfrak{g}}\right)^{\widetilde{\gamma}}\mathfrak{h}^{\widetilde{\psi}} = \mathfrak{g}^{(\widetilde{\alpha}-1)\widetilde{\gamma}}\mathfrak{h}^{\widetilde{\varphi}\widetilde{\gamma}+\widetilde{\psi}}.$$

Using Lemma 5, the exponents of $\mathfrak{g}$ and $\mathfrak{h}$ in the left hand side (i.e., $1 \bmod \mathfrak{q}$ and $0 \bmod \mathfrak{q}$ respectively) and right hand side of this must be equal up to the order of the group, so $1 \equiv (\widetilde{\alpha}-1)\widetilde{\gamma} \bmod \mathfrak{q}$ must hold[6] and therefore $\widetilde{\alpha} \neq 1 \bmod \mathfrak{q}$, as otherwise $\widetilde{\gamma}$ would not exist. Similarly, from the first and third equation of (2) one can conclude that $\widetilde{\alpha} \neq -1 \bmod \mathfrak{q}$.

We next construct a root of ν. From the next two equations (3) and Lemma 6, we can derive that Δc divides $\Delta\alpha$, $\Delta\eta$, $\Delta\epsilon$ and $\Delta\zeta$. Let $\hat{\alpha} = \Delta\alpha/\Delta c$ (i.e., using integer division), and similarly for the other greek letters. Taking the first equation of (3), we get that $C_r = ag^{\hat{\epsilon}}h^{\hat{\zeta}}$ for some a such that $a^2 = 1 \bmod n$ (by Lemma 1). Since $c, c' < p', q'$, by Lemma 3 the value a must be ± 1. Plugging C_r into the second equation of (4) we get

$$1 = a^{\Delta\alpha}g^{\Delta\alpha\hat{\epsilon}}h^{\Delta\alpha\hat{\zeta}}\left(\frac{1}{g}\right)^{\Delta\delta}\left(\frac{1}{h}\right)^{\Delta\beta}.$$

Here $a^{\Delta\alpha}$ must be 1, since if $a^{\Delta\alpha} = -1$ then the product of the other factors in the right hand side of this expression would also have to be -1. But $g, h \in QR_n$ so that that product is also an element of QR_n, while on the other hand $-1 \notin QR_n$. Taking the above expression without $a^{\Delta\alpha}$ in it, then, using Lemma 5 again we can conclude that[7]

[5]The paper is missing the rightmost equality sign in this formula.

[6]The paper erroneously writes q here instead of $\mathfrak{q}$.

[7]The paper erroneously writes $\hat{\beta}$ here instead of $\Delta\beta$. Additionally, for the remainder of the argument to work, β and its cousins with hats and Δ's must be an exponent of h instead of g, so that this identity is $\bmod \operatorname{ord}(h)$ instead of $\bmod \operatorname{ord}(g)$.

$\Delta\beta = \Delta\alpha\hat{\zeta} \bmod \mathrm{ord}(h)$. When put into the first equation of (4), this results in

$$\nu^{\Delta c} = \left(\frac{C_u}{h^{\hat{\zeta}}}\right)^{\Delta\alpha} \qquad \text{which results in} \qquad \nu = b\left(\frac{C_u}{h^{\hat{\zeta}}}\right)^{\hat{\alpha}},$$

with some b such that $b^{\Delta c} = 1$, which must again be ± 1. Actually, if $\hat{\alpha}$ is even then $b = -1$ is not possible since $\nu \in QR_n$, while otherwise we may move b within the brackets of the above expression for ν. Now, without loss of generality we may assume that $\hat{\alpha} > 0$ (otherwise, simply swap s_α with s'_α and similarly for the other responses emitted by the adversary). Then we finally find

$$\nu = u^{\hat{\alpha}} \qquad \text{with} \qquad u = \pm\frac{C_u}{h^{\hat{\zeta}}},$$

where the sign is + if $\hat{\alpha}$ is even. If $\hat{\alpha}$ is odd, then the two $\pm$ possibilities may simply both be tried in order to find the one for which $\nu = u^{\hat{\alpha}}$ holds.

Comparing the definitions of $\widetilde{\alpha}$ and $\hat{\alpha}$, we find that $\hat{\alpha} \bmod \mathfrak{q} = \widetilde{\alpha}$. This completes the proof.[8] $\square$

4 Revocable credentials

In this section, we use this accumulator as part of a credential scheme such as Idemix or BBS+ to add revocation support to that credential scheme. We also discuss what the upper limit B of $X_{A,B} \subset [A, B]$ becomes in such a context.

Denote the attributes of a credential with $m_1, \ldots, m_k$. Without going much detail of either Idemix or BBS+, we note that both of them involve an issuer signature over a Pedersen commitment to the attributes:[9]

- In Idemix, $A = (Z/(S^v \prod_i R_i^{m_i}))^{e^{-1}}$,
- In BBS+, $A = (gh_0^s \prod_i h_i^{m_i})^{(e+x)^{-1}}$.

(In both cases, e is part of the issuer signature and not to be confused with the accumulated primes that we also denote with e.) The verification protocol consists in both cases of the user proving knowledge of

[8] In the paper, the proof concludes with a paragraph containing an analysis on the upper bound of $\hat{\alpha}$ using bounds enforced on s_α. For our use of it in the remainder of this paper we do not need the zero-knowledge proof to deal with such bounds, so we don't include this analysis here, but we do note that this paragraph in the paper contains a formula $\hat{\alpha} = (\Delta\alpha\hat{c} \,\mathrm{rem}\, \mathfrak{q})(\widetilde{\alpha} \,\mathrm{rem}\, \mathfrak{q})$, that should instead be as follows: $\hat{\alpha} = (\Delta\alpha/\Delta c \,\mathrm{rem}\, \mathfrak{q}) = (\widetilde{\alpha} \,\mathrm{rem}\, \mathfrak{q})$.

[9] In fact, this section should work for any credential scheme that is structured like this.

a valid issuer signature over such a commitment, as well as the exponents that that commitment commits to. Therefore, one can combine this with the zero-knowledge proof for $u^e = \nu$ from the previous section as follows:

- The issuer includes e as one of the signed attributes.
- $\mathfrak{C}_e$ is replaced by one of the identities above.
- The zero-knowledge proof from Section 3 is joined with that of the credential scheme for proving knowledge of a valid credential containing e and the attributes.

Note that since the keys for revoking and for issuing a credential are distinct, it is possible to let the tasks of issuance and revoking be done by different parties. In what follows, for ease of terminology and notation we just write "issuer" for both of those parties, but in implementations they may be separated.

We wish to prove security of such a system, given that the credential scheme and accumulator scheme by themselves are secure. By security, we mean that it is not possible to make the verifier accept attributes that have not been issued, *or* attributes that have been issued but revoked.

The zero-knowledge proof of a valid credential and a valid witness (u, e) such that $u^e = \nu$, which lies at the heart of this scheme, guarantees to the verifier only that $e \bmod \mathfrak{q}$ has been signed by the issuer; it provides no guarantees as to the size of e. However, using its signatures over the credentials the issuer can still enforce that only proper numbers $e \in X_{A,B}$ are ever accumulated. As we will see below this is sufficient to prove security, and it allows us to relax the upper limit on B from $B2^{k'+k''+2} < \mathfrak{q}/2$ (as specified in the paper) to the upper limit of the message space of the attributes, i.e., $B < \mathfrak{q}$. This is because the paper uses the factor $2^{k'+k''+2}$ in the zero-knowledge proof to enforce a maximum on the number e of which knowledge is being proven, while in our setup the issuer can enforce that maximum during issuance.

Schematically, we do this as follows. If we have an adversary that can break security in this sense, then as in the proof of security for the zero-knowledge proof (Theorem 2), we can extract from the zero-knowledge proof a valid credential as well as (u, e) such that $e \bmod \mathfrak{q}$ is one of the attributes and $u^e = \nu$. Then an algorithm that uses the adversary in this fashion, simply throws away the credential and returns (u, e), breaks the security of the accumulator. Intuitively, if breaking the accumulator in the sense of coming up with (u, e) such that $u^e = \nu$ is hard without the presence of a credential scheme, then coming up with such (u, e) as well as a valid signature over e is certainly also hard.

Let us make this reduction more formal. As in the existential unforgeability game for signature schemes under adaptive chosen message attacks, the adversary is allowed to query the challenger as much as it wants for a signature over any set of attributes of its choosing. In these queries, the adversary $\mathcal{A}$ is additionally allowed to choose the number e of which it wants a witness u such that $u^e = \nu$, as long as $e \in X_{A,B}$. In response, the adversary obtains a valid credential over the required attributes and e, as well as a witness u for e.

This proof is very similar to the proof of Theorem 2 in the paper, in which an adversary that can break the accumulator if it can choose the numbers e adaptively is reduced to one that does not get to choose them adaptively. The only difference is the addition of a credential scheme, of which the challenger of the adversary holds the private key with which it answers issuance queries of the adversary.

For the set $X_{A,B} \subset [A, B]$ from which the primes e are chosen, we require the following for A and B.

- As the lower bound we take $A > 1$, so that Theorem 1 applies.
- For the upper bound the size limit of the message space of the signature scheme suffices; i.e., $B < \mathfrak{q}$.

Theorem 3. *Let $X_{A,B} \subset [2, \mathfrak{q} - 1]$ be the set from which the primes e are chosen. Under the strong RSA assumption, no probabilistic polynomial algorithm $\mathcal{A}$ exists that with non-negligible probability can convince the verifier that it possesses a valid credential along with a valid witness, whose attributes have not previously been issued in a query,* or *whose witness has either not been added to the accumulator, or removed from it.*

Proof. Suppose that such an adversary $\mathcal{A}$ does exist. We use it to contradict Theorem 1 as follows. Below, we denote with X the set of primes that have occurred in previous queries of the adversary. We proceed as follows.

- First take a random $1 \neq z \in QR_n$, and generate a private/public key pair for the credential scheme.
- When the adversary makes an issuance query for some set of attributes and a number $e \in X_{A,B}$, set u to the current accumulator, compute the updated accumulator as $\nu = u^e$, and add e to X. Thus $\nu = z^X$ (where with exponentiation of z with the set X we mean the successive exponentiation of z with the elements of X). Additionally, create a new credential over the attributes and e. Return the credential and (u, e).

- When the adversary makes a revocation query for e, check that $e \in X$, set the updated accumulator to $\nu = z^{X \setminus \{e\}}$, and remove e from X.
- When the adversary performs the zero-knowledge proof of its credential and witness to a verifier, extract from it the credential and the witness (u, e), and return these.

If the adversary wins, then using the extractor constructed in the previous section we can extract numbers $\hat{e}$ and $\widetilde{e}$ from the adversary such that $\widetilde{e} = \hat{e} \bmod \mathfrak{q}$ and $\nu = u^{\hat{e}}$. The zero-knowledge proof provides no assurances about the size of $\hat{e}$, but this is not necessary: either $\hat{e} < \mathfrak{q}$ so that $\widetilde{e} = \hat{e}$, or not. In the latter case $\hat{e} \notin X$, because the issuer would not have added such an $\hat{e}$ to X. Combined with $u^{\hat{e}} = \nu = z^X$, this constitutes a contradiction with the security of the RSA accumulator as proved in Theorem 1.

Therefore, $\widetilde{e} = \hat{e}$; henceforth we just write e. Then by the unforgeability of the credential scheme, the attributes of the credential including e must correspond to one of the issuance queries. Therefore the number e was added to the accumulator by the challenger, during the issuance query of that credential. That means that the adversary can only win the game by ensuring it is removed from the accumulator using a revocation query for that e before the game ends, which in turn means that $e \notin X$ when the adversary finishes. Again, combined with $u^e = \nu = z^X$ this contradicts the security of the RSA accumulator as proved in Theorem 1. $\square$

5 Instantiations

The proof of knowledge of Section 3 shows not only that the prover knows an accumulated number e such that $e \bmod \mathfrak{q}$ is committed to by $\mathfrak{C}_e = \mathfrak{g}^e \mathfrak{h}^r$, but also that $e \neq \pm 1 \bmod \mathfrak{q}$. If the verifier does not ensure that, then the prover might be able to fool the verifier by using the trivial witnesses $(u, e) = (\nu^{\pm 1}, \pm 1)$, which would indeed satisfy the required identity $\nu = u^e$. The proof achieves this using the two equations for $\mathfrak{g}$ in Equation (1):

$$\mathfrak{g} = \left(\frac{\mathfrak{C}_e}{\mathfrak{g}} \right)^{\gamma} \mathfrak{h}^{\psi} \quad \wedge \quad \mathfrak{g} = (\mathfrak{g}\mathfrak{C}_e)^{\sigma} \mathfrak{h}^{\xi}.$$

As explained in the proof of Theorem 2, the numbers γ, σ are then the inverse modulo $\mathfrak{q}$ of $\alpha \pm 1$, which is therefore unequal to $0 \bmod \mathfrak{q}$.

Here we can provide an optimization using the fact that the numbers e are signed by the issuer. By simply requiring the issuer to never issue a witness (u, e) such that $e = \pm 1 \bmod \mathfrak{q}$, there is no need for the holder of the credential and witness to prove this in the zero-knowledge proof. In the security proof from Section 4 the unforgeability of the credential scheme allows us to conclude that $e \neq \pm 1$, so that the remainder of the proof keeps working.

This means that in anonymous credential scheme implementations these two equations can be omitted from the proof of knowledge of a valid witness, increasing efficiency. The exact same is done in [CL02b] in the example application of the accumulator, in section 4.2.

The above works for any anonymous credential scheme. In the case of Idemix, we can go further. If it is acceptable that only the issuer is able to revoke credentials, we can use the same private key (p, q) to issue Idemix credentials as well as to generate witnesses (u, e). In that case, the group $QR_n \ni u, \nu$ and the group $\mathfrak{G}$ coincide. Referring to the equations being proved by the prover in Equation (1), we can remove all three equations involving $\mathfrak{G}$, and replace $C_e = g^e h^{r_1}$ with the Idemix identity $Z = A^e S^v R_1^{m_1} \cdots R_k^{m_k} R_{k+1}^e$, with which the prover proves knowledge of a valid issuer signature over $(m_1, \ldots, m_k, e)$, since this also constitutes a Pedersen commitment to e. This reduces the equations that the prover has to prove knowledge of from 8 to 4.

6 The RSA-B accumulator

So far, we have considered the accumulator as introduced in the main body of the paper, in which an element e is added and removed to the accumulator by $\nu \mapsto \nu^e$ and $\nu \mapsto \nu^{e^{-1}}$, respectively. In a small remark in Section 4.2 of the paper, it is however remarked that this accumulator can be turned into a different one in which the deletion algorithm is kept the same, but the addition operator does nothing; i.e., $f(\nu, e) = \nu$. It is clear that this is a substantial improvement to the scalability and indeed the feasibility of the system.

However, contrary to the RSA accumulator, the algorithm $f(\nu, e) = \nu$ that updates the accumulator and the algorithm f' that verifies that u is a witness for e by $f'(u, e) = u^e \stackrel{?}{=} \nu$ now no longer coincide. In fact, strictly speaking this means that RSA-B does not completely satisfy the definition from the paper for an accumulator, so we have to review our security proofs to see if they still work.

For this, it is sufficient to require the adversary to commit in advance (before performing the unforgeability game with the challenger) to all of the elements $e_i \in X$ that it is going to use in its queries. In the proof of Theorem 3, one can then in the setup phase construct the accumulator $\nu = z^X$, where X is the set of values committed to by the adversary, and respond to revocation queries as in the proof. In the terminology of [Bal+17] this accumulator is only non-adaptively secure, which is a weaker security notion than the more generally used adaptive security (in which the adversary is allowed to choose the numbers e_i during the unforgeability game, so that it can let them depend on the queries so far).

In this notion of non-adaptive security, however, the adversary still has some control over the numbers e_i that it may use in its queries. In practice, it makes more sense to instead let the issuer decide on these numbers e_i in each issuance query. This removes control over the numbers e_i completely from the adversary. As noted in [Bal+17], this solves the issue: if no adversary exists that can break the RSA accumulator if it has (non-adaptive) control over the e_i, then an adversary without such control over the e_i certainly also cannot exist. Indeed, it is easy to adapt the proof from the previous section to such a setting. In the setup phase, the challenger chooses some primes e_i itself and accumulates them. In an issuance query, the challenger takes one of the e_i, embeds that in a credential and creates a witness for it, and returns that to the adversary. The challenger can then answer revocation queries as in the proof.

We note that in implementations, it is not actually necessary to commit to all of the e_i in advance. Just the fact that such a thing is possible makes the security proof above work, which is sufficient.

Summarizing, in anonymous credential use cases the RSA-B has a significant advantage over the RSA accumulator, while its only downside (the non-adaptive security) is not an issue. Therefore, in implementations RSA-B is preferred.

A Number-theoretic preliminaries

Lemma 1. *Let $n = pq = (2p'+1)(2q'+1)$ be a product of safe primes. Then any element of $\mathbb{Z}_n^*$ can be written as $sr \bmod n$ for some $r \in QR_n$ and s such that $s^2 = 1 \bmod n$.*

Proof. Using the Chinese Remainder theorem (CRT), the fact that $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$ are cyclic, and then CRT again, we have

$$\begin{aligned} \mathbb{Z}_n^* &\cong \mathbb{Z}_p^* \times \mathbb{Z}_q^* \cong \mathbb{Z}_{2p'} \times \mathbb{Z}_{2q'} \\ &\cong \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_{p'} \times \mathbb{Z}_{q'}. \end{aligned} \tag{5}$$

From this we see that if $x \in \mathbb{Z}_n^*$ is arbitrary then $x^{p'q'}$ must be of order 1 or 2; either way it is a square root of 1. Therefore,

$$x = x^{p'q'} x^{p'q'} x \bmod n = x^{p'q'} x^{p'q'+1} \bmod n,$$

in which the first factor is a square root of 1 and the second is a quadratic residue, since $p'q' + 1$ is even. □

From equation (5) we also see that the maximal order that elements of $\mathbb{Z}_n^*$ can attain is $2p'q'$, and that such elements exist, for example $(1, 1, 1, 1)$ under the inverse of the isomorphisms (5). If $x \in \mathbb{Z}_n^*$ is such an element, then $\langle x^2 \rangle$ is a subgroup of $\mathbb{Z}_n^*$ of order $p'q'$. In this subgroup each element has a square root, since $2^{-1} \bmod p'q'$ exists since $\gcd(2, p'q') = 1$. Therefore this group is precisely the subgroup of quadratic residues, which thus has order $p'q'$.

Lemma 2. *Let n be a product of safe primes. If one knows any element $x \in QR_n$ that is not a generator of QR_n, then one can factor n.*

Proof. Suppose x is not a generator; that is, its order is not the maximal order $p'q'$. By Lagrange's theorem, the order of x must divide $\#QR_n = p'q'$. Without loss of generality let its order be p', so that $1 = x^{p'} \bmod n$. Since $n = pq$, reducing modulo q gives the identity

$$1 = x^{p'} \bmod q = (x \bmod q)^{p'} \bmod q.$$

Now $x \bmod q$ is an element of $\mathbb{Z}_q^*$, whose group order is $2q'$, and since x is a quadratic residue the order of $x \bmod q$ cannot be $2q'$, so it must be either 1 or q'. In the latter case, our identity $(x \bmod q)^{p'} = 1 \bmod q$ would imply that q' divides p' which is impossible because p' is prime. Therefore, the order of $x \bmod q$ in $\mathbb{Z}_q^*$ is 1, i.e. $x = 1 \bmod q$. This implies that $x - 1 = aq$ for some a, i.e., $\gcd(n, x - 1) = q$. □

Lemma 3. *Let $n = pq$ be a product of safe primes. Under the Strong RSA assumption, no probabilistic polynomial-time algorithm can compute with non-negligible probability a nontrivial root of $1 \in \mathbb{Z}_n^*$, i.e. a pair (x, m) such that $x^m = 1 \bmod n$, with $\pm 1 \bmod n \neq x \in \mathbb{Z}_n^*$ and $m > 1$.*

Proof. Suppose we have (x, m) such that $x^m = 1 \bmod n$. Note that m is a multiple of the order c of $\langle x \rangle \subset \mathbb{Z}_n^*$. Since $\#\mathbb{Z}_n^* = \phi(n) = (p-1)(q-1) = 4p'q'$, by Lagrange's theorem c can equal only 2, 4, p', or q', or some multiplication of those four numbers. We consider the options.

Suppose $c = 2$. Then $0 \bmod n = x^2 - 1 = (x + 1)(x - 1)$; i.e. for some integer a, $(x+1)(x-1)$ is of the form $(x+1)(x-1) = an = apq$, with $a \neq 0$ since $x \neq \pm 1$. Now since p is prime, it must divide one of the two factors, $x+1$ or $x-1$. Since $x+1 \neq pq = n$ (as we assumed the square root was nontrivial) and $x-1 < n$, it follows that q must divide the other factor. So the factors of n are $\gcd(n, x-1)$ and $\gcd(n, x+1)$.

Next, the proof of Lemma 1 above shows that $\mathbb{Z}_n^*$ has no elements of order $c = 4$ or of an order divided by 4.

If $c = p'$ or $c = q'$ then the previous lemma allows us to factor n using x. The same holds if $c = 2p'$ or $2q'$, because then x^2 will have order p' or q'.

Thus c must be $p'q'$ or $2p'q'$. Going back to m, either way we find that m is a multiple of $p'q'$, i.e., $m = ap'q'$ for some a. Taking any e coprime to m, then, the extended Euclidian algorithm allows us to compute $e^{-1} \bmod m = e^{-1} \bmod ap'q'$, which reduces to $e^{-1} \bmod p'q'$. Given some $y \in QR_n$ that means that $u = y^{e^{-1}}$ is a root of y, so that (u, e) is a Strong RSA instance for y. □

Lemma 4. *Let $n = (2p' + 1)(2q' + 1)$ be a product of safe primes. Under the Strong RSA assumption, no probabilistic polynomial-time algorithm can compute with non-negligible probability a number d such that* $\gcd(d, p'q') \neq 1$.

Proof. If $\gcd(d, p'q') \neq 1$ then the number d must be a multiple of p' or q' (or both). Without loss of generality we may assume $d = ap'$ for some number a. Taking any $1 \neq w \in QR_n$, then by Lagrange's theorem we find that the order of the element $w^d = w^{ap'}$ must be one of the following:

- q', which is impossible by Lemma 2,
- 1, meaning that $w^d = 1 \bmod n$ which is impossible by Lemma 3.

□

Lemma 5. *Let G be a group in which the discrete logarithm problem holds. Then no probabilistic polynomial-time algorithm can, on input $(g_1, \ldots, g_k)$ where the g_i are randomly generated, compute with non-negligible probability numbers $a_1, \ldots, a_k$ satisfying*

$$g_1^{a_1} g_2^{a_2} \ldots g_n^{a_k} = 1 \in G.$$

When one encounters such an expression, this allows us to conclude that with overwhelming probability $a_1 = \cdots = a_k = 0$. We will not prove this here, but see e.g. [Bra00, p. 60].

Lemma 6. *Under the strong RSA assumption, given a modulus n along with random elements $g, h \in QR_n$, no probabilistic polynomial-time algorithm can compute with non-negligible probability an element $w \in \mathbb{Z}_n^*$ and integers a, b, c such that*

$$w^c = g^a h^b \bmod n \quad \textit{and } c \textit{ does not divide } a \textit{ or } b.$$

When one encounters such an expression, this allows us to conclude that with overwhelming probability, c divides a and b. For a proof, see [CS03].

References

[ASM06] M. H. Au, W. Susilo, and Y. Mu. "Constant-Size Dynamic k-TAA". In: *Security and Cryptography for Networks*. Ed. by R. De Prisco and M. Yung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 111–125. ISBN: 978-3-540-38081-8.

[Bal+17] F. Baldimtsi, J. Camenisch, M. Dubovitskaya, A. Lysyanskaya, L. Reyzin, K. Samelin, and S. Yakoubov. "Accumulators with Applications to Anonymity-Preserving Revocation". In: *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. `https://eprint.iacr.org/2017/043`. 2017, pp. 301–315. DOI: `10.1109/EuroSP.2017.13`.

[Bra00] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000.

[CDL16] J. Camenisch, M. Drijvers, and A. Lehmann. "Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited". In: *Trust and Trustworthy Computing*. Ed. by M. Franz and P. Papadimitratos. `https://eprint.iacr.org/2016/663`. Cham: Springer International Publishing, 2016, pp. 1–20. ISBN: 978-3-319-45572-3.

[CL02a] J. Camenisch and A. Lysyanskaya. "A Signature Scheme with Efficient Protocols". In: *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*. Ed. by S. Cimato, C. Galdi, and G. Persiano. Vol. 2576. Lecture Notes in Computer Science. Springer, 2002, pp. 268–289.

[CL02b] J. Camenisch and A. Lysyanskaya. "Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials". In: *Advances in Cryptology — CRYPTO 2002*. Ed. by M. Yung. `https://cs.brown.edu/people/alysyans/papers/camlys02.pdf`. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 61–76. ISBN: 978-3-540-45708-4.

[CS03] J. Camenisch and V. Shoup. "Practical Verifiable Encryption and Decryption of Discrete Logarithms". In: *Advances in Cryptology - CRYPTO 2003*. Ed. by D. Boneh. `https://www.shoup.net/papers/verenc.pdf`. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 126–144. ISBN: 978-3-540-45146-4.

[IBM12] IBM Research Zürich Security Team. *Specification of the Identity Mixer Cryptographic Library, version 2.3.4*. Tech. rep. IBM Research, Zürich, Feb. 2012.