Secure Accumulators from Euclidean Rings without Trusted Setup

Helger Lipmaa

Institute of Computer Science, University of Tartu, Estonia

Abstract. Cryptographic accumulators are well-known to be useful in many situations. However, the most efficient accumulator (the RSA accumulator) it is not secure against a certificate authority who has herself selected the RSA modulus n. We generalize previous work and define the root accumulator in modules over Euclidean rings. We prove that the root accumulator is secure under two different pairs of assumptions on the module family and on the used hash function. Finally, we propose a new instantiation of the root accumulator, based on class groups of imaginary quadratic order, that combines the best properties of previous solutions. It has short (non)membership proofs like the RSA accumulator, and at the same time it is secure against a malicious certificate authority. Up to this point, this seems to be the only unique application of class groups of imaginary quadratic orders, and we hope that this paper will motivate more research on cryptography in the said groups.

Keywords: Class groups of imaginary quadratic order, cryptographic accumulators, Euclidean rings.

1 Introduction

Cryptographic accumulators have been proven to be extremely useful in the public-key infrastructure, anonymous credential systems and many other applications. Briefly, in a cryptographic accumulator, for any document set S, a server can compute a short digest $\mathsf{Dig}(S)$, such that for any candidate document m one can find a succinct (non)membership proof $\mathsf{Proof}(m, S)$ of m (not) belonging to S. The digest $\mathsf{Dig}(S)$ is published, and everybody can obtain it in an authenticated manner. Finally, different clients use the verification algorithm Ver. It is required that $\mathsf{Ver}(m, \mathsf{Dig}(S), \mathsf{Proof}(m, S)) = \mathsf{Member}$ if $m \in S$, and (in some of the papers like [5,6,23]) $\mathsf{Ver}(m, \mathsf{Dig}(S), \mathsf{Proof}(m, S)) = \mathsf{NotMember}$ if $m \notin S$. Accumulators are required to be collision-resistant, that is, it should be difficult to construct a triple (m, S, p), such that $m \notin S$ but $\mathsf{Ver}(m, \mathsf{Dig}(S), p) = \mathsf{Member}$ [1].

One can construct collision-resistant accumulators (with nonmembership proofs) based on hash-trees, see [5,6]. However, hash-tree based solutions have relatively long — logarithmic in $|S|$ — (non)membership proofs. The more succinct RSA accumulator was introduced in [2], further studied in [1,25,26,11], and proven to be collision-resistant in [1]. Further accumulators have been proposed in say [24,10].

F. Bao, P. Samarati, and J. Zhou (Eds.): ACNS 2012, LNCS 7341, pp. 224–240, 2012.

Unfortunately, one cannot rely on the accumulating party (say, the certificate authority) to honestly generate the value $\mathsf{Dig}(S)$.[1] In particular, she could publish d (not necessarily knowing the corresponding S) such that she can later generate both membership and nonmembership proofs for some selected elements m. To tackle this situation, Buldas, Laud and Lipmaa [5,6] required accumulators to be *undeniable* in the next sense: it should be infeasible to generate a tuple $(m, d, p, \overline{p})$, such that $\mathsf{Ver}(m, d, p) = \mathsf{Member}$ but $\mathsf{Ver}(m, d, \overline{p}) = \mathsf{NotMember}$. (The same security requirement — under different names — has been independently reinvented in say [9].) Thus, in the case of certificate management, when a client sees a certificate m, digest d and (say) a proof p that m was revoked, she can be certain that there does not exist a contradictory proof $\overline{p}$ that m was not revoked. Buldas, Laud and Lipmaa also constructed a concrete undeniable accumulator based on hashed search trees. (They called it an undeniable attester since it is not based on the RSA accumulator.) Because their solution is based on hashed search trees, it is trapdoorless and thus secure against a malicious server. Unfortunately, there the (non)membership proofs p have length that is logarithmic in the size of S.

For a long time, it was not known how to construct short nonmembership proofs for the RSA accumulators. Only in 2007, Li, Li and Xue [23] showed how to do that. In their modification to the RSA accumulator, a membership proof consists of one group element and a nonmembership proof consists of one group element and one exponent. Unfortunately, in the case of the RSA accumulator, the server can generate the RSA modulus n herself, and thus knowing the factorization of n she can efficiently break the accumulator. That means that the Li-Li-Xue construction is only secure in the trusted setup model where n is generated by a trusted third party who does not disclose its factorization to the server. Sander [25] tried to eliminate the trapdoor in the RSA accumulator but his construction, while trapdoorless, is very inefficient. Moreover, from the perspective of a client who just started to use the accumulator, it still does not guarantee that the server does not know the trapdoor. Our goal is to get rid of the trusted setup assumption, and to achieve efficiency that is comparable to that of the RSA accumulator.

Our Contributions. We first substantially generalize the RSA accumulator as modified by Li, Li and Xue. The generalized *root accumulator* works in $\mathcal{R}_{\mathcal{D}}$, which is a family of modules D over Euclidean rings R, and uses a hash function (family) H. This generalization serves two different purposes. First, by generalizing the algebraic setting to the widest one, it may become possible in the future to find other more efficient instantiations of the primitive. (Even if at this

[1] The original motivation of this line of research is digital time stamping, where the digest over answers to time-stamping queries is computed by the time-stamping authority [17,7,8]. Cryptographic methods are precisely in place to counter the case where the authority may be malicious. In particular, a malicious time-stamping authority can clearly compute a spurious value of $\mathsf{Dig}(S)$. See [5] for more discussion and motivation.

moment, the only known instantiations consist of Abelian groups D and $R = \mathbb{Z}$, with the module operation $\circ : R \times D \to D$ defined as $\alpha \circ x := x^{\alpha}$.) Second, the construction of the root accumulator depends crucially on the existence of the Extended Euclidean Algorithm in the underlying ring. In addition, most of the security reductions of this paper make an explicit use of the Extended Euclidean Algorithm. Thus, we think it is methodologically useful to explicitly point out that the Extend Euclidean Algorithm algorithm must exist in the underlying algebraic structure, and must be efficient. While modules over rings have been used in cryptography before, see [16], we are unaware of any previous use of modules over Euclidean rings in cryptography. Thus, this generalization may be a contribution by itself.

Before proving the security of the root accumulator, we must define the corresponding security notions and underlying security assumptions. The first technical difficulty (and novelty) there is that because we want the accumulator to be secure without trusted setup, the security definitions will become more involved. In particular, an accumulator must have a public key divided into two parts, one of which (say, the RSA modulus n) is generated by using a public randomness known by the adversary, and another one (say, a generator of a large subgroup in $\mathbb{Z}_n^*$) can be chosen by using a non-public randomness. Because it was the trapdoor in n that we were worried about, this division is fine for our purposes. (We leave it as an interesting open question to solve the second part in an accountable way.) Similarly, when defining the security assumptions, we must consider the case where the adversary knows the randomness that is used when choosing the module (again, in the case of the RSA accumulator this corresponds to the adversary knowing the factorization of n) where the root accumulator will be run.

Then, we show that the root accumulator is both collision-resistant and undeniable if either (a) $\mathcal{R}_{\mathcal{D}}$ is a *strong prime root module family* and H is a *prime-valued injective function [1]*, or (b) $\mathcal{R}_{\mathcal{D}}$ is a *strong divisible root module family [13]* and H is a *division-intractable function family [15]*. (Corresponding security definitions are given later in the paper.)

Based on those results, we show that if factorization is hard in the Euclidean ring, then the security of the root accumulator—given that H is prime-valued injective—is based on a presumably weaker assumption than the strong root assumption (e.g., the security of the RSA accumulator is based on a presumably weaker assumption than the strong RSA assumption). We also show that the strong divisible root assumption is equivalent to the strong root assumption (which is known to be secure in the generic group model [14]), given that the module satisfies another seemingly unrelated *small root assumption*. (The latter is related but generalizes significantly the small root assumption of [13].)

As a concrete instantiation, we propose to use class groups of imaginary quadratic orders with a large discriminant Δ where $-\Delta$ is a prime [4]. Many previous cryptographic schemes are based on the strong root assumption in such groups. Importantly, Δ can be chosen by a malicious adversary with only negligibly changing her probability of breaking the root accumulator. While the

applicability of class groups of imaginary quadratic order has been studied quite extensively in the cryptographic literature (see [3,18] for an overview), one has been mostly interested in such groups because they are one of the very few group families known (in addition to say multiplicative groups of residue rings and (hyper)elliptic curve groups) that are suitable for cryptographic use. We show that there is a natural cryptographic problem—construction of secure accumulators without trusted setup—for which class groups of imaginary quadratic order are the *only* known suitable group family. We hope that this will generate additional interest in cryptography based on such groups.

Basic Notation. We assume that Member, NotMember and Error are special symbols. k denotes the security parameter. The working time of all algorithms and the security of all primitives is measured as a function of the security parameter k. $\mathrm{negl}(k)$ denotes an arbitrary negligible function in k, $\mathrm{poly}(k)$ denotes an arbitrary polynomial function in k. PPT means probabilistic polynomial time. We note that in the context of this paper, the adversary can always be non-uniform; however, our reductions themselves are all uniform. If S is a set, then $x \leftarrow S$ denotes random sampling, and $x \leftarrow S(\omega)$ denotes random sampling while using ω as the random tape. If A is an algorithm, then $x \leftarrow A(y)$ denotes random sampling of the output of A, given input y.

2 Collision-Resistant and Undeniable Accumulators

First, we will state the syntax of accumulators that allow nonmembership proofs as in [5,6,23]. (In [5,6], an accumulator with nonmembership proofs was called an *attester*.) Informally, an accumulator is a mechanism that for each candidate element m and a set S produces a succinct (non)membership proof that attests to the fact that $m \in S$ or $m \notin S$. Based on m, the short digest of S and the corresponding proof (and without access to any other information), one can later verify whether $m \in S$ or not.

Definition 1 (Accumulator). *Let M, D and P be three sets (the message set, the digest set and the proof set correspondingly). A quadruple* $\mathsf{Acc} = (\mathsf{Gen}, \mathsf{Proof}, \mathsf{Dig}, \mathsf{Ver})$ *of PPT algorithms is a* (strong) accumulator, *if it satisfies the next conditions:*

Generating algorithm $\mathsf{Gen}(1^k)$ *outputs a public key* pk.
Membership algorithm $\mathsf{Proof}_{\mathsf{pk}}(m, S)$**:** *If $m \in M$ and $S \subseteq M$, then it outputs a* membership proof *$p \in P$, otherwise it outputs* Error.
Digest algorithm $\mathsf{Dig}_{\mathsf{pk}}(S)$**:** *If $S \subseteq M$ then it outputs a* digest *$d \in D$, otherwise it outputs* Error.
Verification algorithm $\mathsf{Ver}_{\mathsf{pk}}(m, d, p)$**:** *If $m \in M$, $d \in D$ and $p \in P$ then it outputs either* Member *or* NotMember, *otherwise it outputs* Error.

An accumulator must satisfy the next correctness property: for valid $\mathsf{pk} \in \mathsf{Gen}(1^k)$, *$m \in M$ and $S \subseteq M$,* $\mathsf{Ver}_{\mathsf{pk}}(m, \mathsf{Dig}_{\mathsf{pk}}(S), \mathsf{Proof}_{\mathsf{pk}}(m, S))$ *outputs* Member *if $m \in S$, and* NotMember *if $m \notin S$.* □

Note that because all algorithms work in probabilistic polynomial time, it is always implicitly required that $|S| = \mathrm{poly}(k)$.

Definition 2 (Security in Trusted Setup Model). *Let* $\mathsf{Acc} = (\mathsf{Gen}, \mathsf{Proof}, \mathsf{Dig}, \mathsf{Ver})$ *be an accumulator.* Acc *is* collision-resistant [1] *(in the trusted setup model) if*

$$\Pr\left[\begin{array}{l}\mathsf{pk} \leftarrow \mathsf{Gen}(1^k), (m, S, p) \leftarrow A(\mathsf{pk}) : \\ m \notin S \wedge \mathsf{Ver}_{\mathsf{pk}}(m, \mathsf{Dig}_{\mathsf{pk}}(S), p) = \mathsf{Member}\end{array}\right] = \mathrm{negl}(k)$$

for any PPT adversary A. Acc *is* undeniable [5,6,23] *(in the trusted setup model) if*

$$\Pr\left[\begin{array}{l}\mathsf{pk} \leftarrow \mathsf{Gen}(1^k), (m, d, p, \overline{p}) \leftarrow A(\mathsf{pk}) : \\ \mathsf{Ver}_{\mathsf{pk}}(m, d, p) = \mathsf{Member} \wedge \mathsf{Ver}_{\mathsf{pk}}(m, d, \overline{p}) = \mathsf{NotMember}\end{array}\right] = \mathrm{negl}(k)$$

for any PPT adversary A.

Proofs p and $\overline{p}$ are *contradictory* if for some m and d, $\mathsf{Ver}_{\mathsf{pk}}(m, d, p) = \mathsf{Member}$ and $\mathsf{Ver}_{\mathsf{pk}}(m, d, \overline{p}) = \mathsf{NotMember}$.

It was proven in [1] that the RSA accumulator [2] is collision-resistant (in the trusted setup model). However, in several potential usage scenarios of accumulators, the trusted setup assumption is really inappropriate. For example, imagine the setting (similar to digital time stamping [17,7], where cryptographic methods are introduced precisely to obtain security against a corrupt authority) where a certificate authority periodically revokes certificates. Instead of periodically publishing certificate revocation lists, she publishes their short digests. To every client who wants to check whether or not some particular certificate was revoked during that period, she also sends a succinct (non)membership proof with respect to this revocation list. If the accumulator is undeniable (without any trusted setup), the client can be certain that nobody else has a contradictory proof.

In the case of the RSA accumulator, the certificate authority may know the factorization $n = PQ$ of the RSA modulus n. (The same attack is also valid in other scenarios, obviously.) Even when using threshold methods to generate n, there is always a coalition of parties who know P and Q. In striking contrast with many other cryptographic applications, here we cannot assume that the client herself participated in the generation of n, since she might have not been using the services of this certificate authority at the that time. See [5,6] for more motivation and [25] for an early paper on trapdoorless RSA accumulator.

We will now define security in the case without trusted setup. The RSA accumulator is not secure without trusted setup, because even a semi-honest party who generates n can later cheat (e.g., by revealing its prime factors after being adaptively corrupted). We tackle this problem by introducing a new Setup algorithm (that generates the algebraic structure we are working in), and requiring that the adversary must have access to the random tape ω of Setup. On the other

hand, Gen's random tape must remain hidden. (In the case of RSA accumulator, the latter corresponds to the part that is used while generating a generator of some large subgroup of $\mathbb{Z}_n^*$.) We thus naturally augment the definition of accumulators with the Setup algorithm, and assume that all other algorithms get the output of Setup as one of the inputs.

Definition 3 (Security without Trusted Setup). *Let* $\mathsf{Acc} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Proof}, \mathsf{Dig}, \mathsf{Ver})$ *be an accumulator.* Acc *is* collision-resistant *(without trusted setup) if*

$$\Pr_{\omega}\left[\begin{array}{l}\mathsf{parm} \leftarrow \mathsf{Setup}(1^k, \omega), \mathsf{pk} \leftarrow \mathsf{Gen}(1^k, \mathsf{parm}),\\ (m, S, p) \leftarrow A(\omega, \mathsf{parm}, \mathsf{pk}) :\\ m \notin S \wedge \mathsf{Ver}_{\mathsf{pk},\mathsf{pk}}(m, \mathsf{Dig}_{\mathsf{parm},\mathsf{pk}}(S), p) = \mathsf{Member}\end{array}\right] = \mathrm{negl}(k)$$

for any PPT adversary A. Acc *is an* undeniable accumulator *(without trusted setup) if*

$$\Pr_{\omega}\left[\begin{array}{l}\mathsf{parm} \leftarrow \mathsf{Setup}(1^k, \omega), \mathsf{pk} \leftarrow \mathsf{Gen}(1^k, \mathsf{parm}),\\ (m, d, p, \overline{p}) \leftarrow A(\omega, \mathsf{parm}, \mathsf{pk}) :\\ (\mathsf{Ver}_{\mathsf{parm},\mathsf{pk}}(m, d, p) = \mathsf{Member}) \wedge\\ (\mathsf{Ver}_{\mathsf{parm},\mathsf{pk}}(m, d, \overline{p}) = \mathsf{NotMember})\end{array}\right] = \mathrm{negl}(k)$$

for any PPT adversary A.

Note that this is somewhat similar to security definitions in the common reference string model, where parm is honestly chosen, and the adversary (usually a simulator) can choose parm herself together with a corresponding trapdoor ω. In fact, one can consider a stronger requirement, where ω is not only known to the adversary but actually chosen by her. However, in this case in all subsequent security assumptions one would have to assume that the assumptions hold even if the adversary can choose the underlying module. Unfortunately, no module families are known where such security assumptions would hold. On the positive side, checking that pk is generated correctly is more plausible than checking that the trapdoor information ω is not known to the adversary. For example, pk can be generated by using verifiable randomness published in newspapers or the NIST beacon (`http://www.nist.gov/itl/csd/ct/nist_beacon.cfm`).

3 Module-Based Cryptography

Many public-key primitives are based on groups. We generalize the group-based setting to the module-based one. For this we generalize several well-known notions and introduce a few new ones. In the next section, we propose an accumulator that is based on a module over a Euclidean ring. Within this paper, all rings are commutative.

Algebraic Background and Definitions. A (left) R-module over the ring R consists of an Abelian group $(D,+)$ and an operation $R \times D \to D$ (that we denote by $\alpha \circ g$), such that for all $\alpha, \beta \in R$ and $x, y \in D$, we have (a) $\alpha \circ (x+y) = \alpha \circ x + \alpha \circ y$, (b) $(\alpha+\beta) \circ x = \alpha \circ x + \beta \circ x$, (c) $(\alpha \cdot \beta) \circ x = \alpha \circ (\beta \circ x)$, and (d) $1 \circ x = x$.

A commutative ring R with identity is called an *integral domain* if for all $\alpha, \beta \in R$, $\alpha\beta = 0$ implies $\alpha = 0$ or $\beta = 0$. A ring R is *Euclidean* if it is an integral domain and there exists a function $\deg : R \to \mathbb{Z}^+$, called the Euclidean degree, such that (a) if $\alpha, \beta \in R$ with $\alpha\beta \neq 0$ and $\alpha \neq 0$, then $\deg(\alpha) \leq \deg(\alpha\beta)$ and (b) if $\alpha, \beta \in R$ then there exist $\gamma, \delta \in R$ such that $\alpha = \gamma\beta + \delta$ with either $\delta = 0$, or $\delta \neq 0$ and $\deg(\delta) < \deg(\beta)$. Every Euclidean ring possesses a multiplicative identity. An element α of R which is neither 0 nor 1 is called *irreducible* if there are no non-1 elements β and γ with $\alpha = \beta \cdot \gamma$. Define $\textsc{Irr}(R)$ to be the set of irreducible elements of R.

Some examples of Euclidean rings R are $\mathbb{Z}$ with $\deg(\alpha) := |\alpha|$, $\mathbb{Z}[i]$ (the ring of Gaussian integers) with $\deg(\alpha) := |\alpha|^2$, $K[X]$ for arbitrary field K with $\deg(\alpha)$ being the degree of polynomial α when $\alpha \neq 0$, the ideals of polynomial ring $\mathbb{Z}_p[X]$ (that are modules over $\mathbb{Z}_p[X]$), and arbitrary field K where $\deg(\alpha) := 1$ when $\alpha \neq 0$. An example of a non-commutative Euclidean ring is the polynomial ring $P[x]$ over a skew field (division ring) P. In all such cases one can talk about the irreducible elements of R.

Intractable Problems in Modules. Because we want the accumulator to be secure without trusted setup, it must also be the case that in the underlying security assumptions the adversary can see the coins used while selecting the concrete module.

Definition 4 (Security Assumptions without Trusted Setup). *Let $\mathcal{R}_{\mathcal{D}} = ((R_i)_{D_i})$ be a family of modules with $i \in I$ and an efficient deterministic algorithm $\mathsf{Setup}(1^k, \omega)$ that picks some $i \in I$. We assume that A is a stateful algorithm.*

1. *$\mathcal{R}_{\mathcal{D}}$ is a* discrete logarithm module family *if for every PPT adversary A,*

$$\Pr_{\omega}[R_D \leftarrow \mathsf{Setup}(1^k, \omega), (x, y) \leftarrow D, \alpha \leftarrow A(x, y, \omega) : \alpha \circ y = x] = \mathrm{negl}(k) \ .$$

2. *$\mathcal{R}_{\mathcal{D}}$ is an* order module family *if for every PPT adversary A,*

$$\Pr_{\omega}[R_D \leftarrow \mathsf{Setup}(1^k, \omega), x \leftarrow D, y \leftarrow A(x, \omega) : \mathrm{ord}(x) = y] = \mathrm{negl}(k) \ .$$

3. *$\mathcal{R}_{\mathcal{D}}$ is a* root module *if for every PPT adversary A,*

$$\Pr_{\omega}[R_D \leftarrow \mathsf{Setup}(1^k, \omega), x \leftarrow D, \alpha \leftarrow R, y \leftarrow A(x, \alpha, \omega) : \alpha \circ y = x] = \mathrm{negl}(k) \ .$$

4. *$\mathcal{R}_{\mathcal{D}}$ is a* strong prime root module *if for every PPT adversary A,*

$$\Pr_{\omega}\left[\begin{array}{l} R_D \leftarrow \mathsf{Setup}(1^k, \omega), x \leftarrow D, (y, \alpha) \leftarrow A(x, \omega) : \\ (\alpha \circ y = x) \wedge (\alpha \in \textsc{Irr}(R)) \end{array}\right] = \mathrm{negl}(k) \ .$$

Setup Algorithm $\mathsf{Setup}(1^k, \omega)$**:**
Generate random $i \leftarrow I$ according to ω. Let $\mathsf{parm} \leftarrow i$.

Generating Algorithm $\mathsf{Gen}(1^k, \mathsf{parm})$**:**
Generate random $g \leftarrow D_i$. Publish $\mathsf{pk} \leftarrow g$.

Digest Algorithm $\mathsf{Dig}_{\mathsf{parm},\mathsf{pk}}(S)$**:**
1. If $i \notin I$ or $g \notin D_i$, then return Error.
2. If $S \not\subseteq M_i$, then output Error.
3. Otherwise, output $(\prod_{s \in S} H(s)) \circ g$.

(Non)Membership Proof Algorithm $\mathsf{Proof}_{\mathsf{parm},\mathsf{pk}}(m, S)$**:**
1. If $i \notin I$ or $g \notin D_i$, then return Error.
2. If $m \notin M_i$ or $S \not\subseteq M_i$, then return Error.
3. If $m \in S$ then define $\mathsf{Proof}_{\mathsf{pk}}(m, S) := (\prod_{s \in S \setminus \{m\}} H(s)) \circ g$.
4. Otherwise, let $\delta \leftarrow \prod_{s \in S} H(s) \in R_i$. Because R_i is Euclidean and $\gcd(H(m), \delta) = 1$, there exist $\alpha, \beta \in R_i$, such that $\alpha \cdot H(m) + \beta \cdot \delta = 1$. Let $\mathsf{Proof}_{\mathsf{pk}}(m, S) := (\alpha \circ g, \beta)$.

Verification $\mathsf{Ver}_{\mathsf{parm},\mathsf{pk}}(m, d, p)$**:**
1. If $i \notin I$ or $g \notin D_i$, then return Error.
2. If $m \notin M_i$ or $d \notin D_i$, then return Error.
3. If $p \in D_i$, then check whether $H(m) \circ p = d$.
   If it is, then return Member, else return Error.
4. Otherwise, if $p = (q, \beta) \in D_i \times R_i$, then check whether $H(m) \circ q + \beta \circ d = g$.
   If it is, then return NotMember, else return Error.
5. Otherwise, return Error.

Fig. 1. Root accumulator for $(\mathcal{R}_{\mathcal{D}}, H)$

5. $\mathcal{R}_{\mathcal{D}}$ *is a* strong root module *if*

$$\Pr_{\omega}\left[\begin{array}{l} R_D \leftarrow \mathsf{Setup}(1^k, \omega), x \leftarrow D, (y, \alpha) \leftarrow A(x, \omega) : \\ (\alpha \circ y = x) \wedge (\alpha \neq 1) \end{array}\right] = \mathrm{negl}(k)$$

   for every PPT adversary A.
6. $\mathcal{R}_{\mathcal{D}}$ *is a* strong divisible root module *if*

$$\Pr_{\omega}\left[\begin{array}{l} R_D \leftarrow \mathsf{Setup}(1^k, \omega), x \leftarrow D, (y, \alpha, \beta) \leftarrow A(x, \omega) : \\ ((\alpha\beta) \circ y = \beta \circ x) \wedge (\alpha \neq 1) \end{array}\right] = \mathrm{negl}(k)$$

   for every PPT adversary A.

In the trusted setup model, one does not require security in the case A knows ω*, and thus A may be able to break the assumption by obtaining access to it.*

The strong prime root and (to certain extent) the strong divisible root assumption are novel, while others are generalizations of well-known assumptions. The assumptions are ordered starting from the "weakest" one, see Sect. 6. For example, if one can solve the discrete logarithm problem in R_D then one also clearly solve the order problem.

The cryptographically familiar example of modules is R_D with $\alpha \circ x := x^\alpha$ for $x \in D$ and $\alpha \in R = \mathbb{Z}$. The order assumption for groups is well known—for the RSA group, it is also called the RSA assumption. That RSA groups are strong root groups was postulated in [1] (the corresponding assumption being called the strong RSA assumption). Damgård and Fujisaki [13] enlisted some candidate strong divisible root groups. Note that in the case of RSA groups the assumptions can only hold in the trusted setup model.

4 Accumulator with Prime-Valued Injective Functions

In this section, we propose the root accumulator for R-modules that generalizes previous work of [23] that considered only the setting of RSA groups, and prime inputs m.

Setting. For some set M and Euclidean ring R, function $f : D \to R$ is a *prime-valued injective function* if it is an injective function $D \to \textsc{Irr}(R)$. We will not propose new prime-valued injective functions, see [1,15] for some existing designs. Let $\mathcal{R}_{\mathcal{D}} = (R_i)_{D_i}$ for $i \in I$, where D_i is an Abelian group and R_i is a Euclidean ring. Let H be a prime-valued injective function $H : M_i \to R_i$. Here, I depends on k and H depends on i. The root accumulator is depicted by Fig. 1.

Security Proofs.

Theorem 1. *The root accumulator satisfies the correctness property.*

Proof. Assume all participants are honest. Thus $i \in I$, $g \in D_i$, $m \in M_i$ and $d = \mathsf{Dig}_{\mathsf{pk}}(S) \in D_i$. We need to show that if $m \in S$ then $\mathsf{Ver}_{\mathsf{pk}}(m, d, p) = \mathsf{Member}$, and if $m \notin S$ then $\mathsf{Ver}_{\mathsf{pk}}(m, d, p) = \mathsf{NotMember}$. First, if $m \in S$ then $p = (\prod_{s \in S \setminus \{m\}} H(s)) \circ g$. Thus,

$$H(m) \circ p = H(m) \circ ((\prod_{s \in S \setminus \{m\}} H(s)) \circ g) = (\prod_{s \in S} H(s)) \circ g = d \ .$$

Second, if $m \notin S$ then $p = (q, \beta) \in D_i \times R_i$, with $q = \alpha \circ g$ and $\alpha \cdot H(m) + \beta \cdot \delta = 1$ for some α. But then

$$H(m) \circ q + \beta \circ d = (\alpha \cdot H(m)) \circ g + (\beta \cdot \delta) \circ g = 1 \circ g = g \ ,$$

since $\delta = \prod_{s \in S} H(s)$ and $d = \delta \circ g$. □

The next two proofs show that in some sense, collision-resistancy and undeniability of the root accumulator are equivalent, though their reductions to the same underlying problem have different costs. In general, it seems to be difficult to prove that every undeniable accumulator is collision-resistant, because in the case of undeniability the adversary has to return a (possibly fake) digest, while in the case of the collision-resistancy, the adversary has to return a set which may not be easily computable from the fake digest. Moreover, clearly not every collision-resistant accumulator is undeniable; see [5,6] for discussions.

Theorem 2 (Sufficient Conditions (with Prime-Valued Injective Functions)). *Let H be a prime-valued injective function. (1) If $\mathcal{R}_{\mathcal{D}}$ is a strong prime root module family, then the root accumulator is collision-resistant. (2) If $\mathcal{R}_{\mathcal{D}}$ is a strong prime root module family, then the strong root accumulator is undeniable.*

Proof. (1) Construct a machine B to break the strong prime root assumption using as the oracle an adversary A that breaks the collision-resistancy of the root accumulator.

1. B obtains $i \leftarrow \mathsf{Setup}(1^k, \omega)$, and random ω.
2. B obtains his challenge $x \leftarrow D_i$.
3. B queries $(m, S, p) \leftarrow A(x, \omega)$.
4. B sets $\delta \leftarrow \prod_{s \in S} H(s)$, $d \leftarrow \delta \circ x$. B finds a pair (α, β), such that $\alpha \cdot H(m) + \beta \cdot \delta = 1$.
5. B returns $(\alpha \circ x + \beta \circ p, H(m))$.

Since $m \notin S$ and H is prime-valued injective, thus $\gcd(H(m), \delta) = 1$. Because R_i is a Euclidean ring, (α, β) can be found efficiently by using the Extended Euclidean Algorithm. Then

$$\begin{aligned} H(m) \circ (\alpha \circ x + \beta \circ p) &= (\alpha \cdot H(m)) \circ x + (\beta \cdot H(m)) \circ p \\ &= (\alpha \cdot H(m)) \circ x + (\beta \cdot \delta) \circ x \\ &= (\alpha \cdot H(m) + \beta \cdot \delta) \circ x = x \ . \end{aligned}$$

Clearly if A is successful then B is successful. B's running time is dominated by the running time of A and by the time it takes to execute the Extended Euclidean algorithm (and thus, R_i has to be Euclidean).

(2) Construct a machine B to break the strong root assumption using as the oracle an adversary A that breaks the undeniability of the root accumulator.

1. B obtains $i \leftarrow \mathsf{Setup}(1^k, \omega)$, and random ω.
2. B obtains his challenge $x \leftarrow D_i$. B sets $\mathsf{pk} = (i, x)$.
3. B queries $(m, d, p, \overline{p}) \leftarrow A(\mathsf{pk}, \omega)$, where $\overline{p} = (q, \alpha)$.
4. B returns $(q + \alpha \circ p, H(m))$.

If A is successful, then $H(m) \circ p = d$ and $H(m) \circ q + \alpha \circ d = x$. Thus,

$$H(m) \circ (q + \alpha \circ p) = H(m) \circ q + (\alpha \cdot H(m)) \circ p = x \ .$$

Therefore, B breaks the strong root problem with the same probability that A breaks the undeniability of the root accumulator, in time that is dominated by A's running time. (For this reduction to go through, R_i does not have to be Euclidean.) □

5 Accumulator with Division-Intractable Function Family

One of the drawbacks of the root accumulator (as described in the previous section) is that prime-valued injective functions (see [1] for some examples) may

be inefficient. In this section, we consider a variation of the root accumulator that works with potentially more efficient division-intractable functions [15]. However, due to that, it is based on a (probably) stricter assumption on $\mathcal{R}_{\mathcal{D}}$.

Division-intractable functions. Let I be an index set. As always, let R_D be an R-module over Euclidean ring R. A hash function family $\mathcal{H} = \{\mathcal{H}_i\}$ with $H : M_i \to R_i$ for every $H_i \in \mathcal{H}_i$ is a *division-intractable function family* [15] if

$$\Pr\left[\begin{array}{l} H_i \leftarrow \mathcal{H}_i, (m, S) \leftarrow A(H_i) : \\ (S \subseteq D_i) \wedge (m \in D_i \setminus S) \wedge (H(m) \mid \prod_{s \in S} H(s)) \end{array}\right] = \mathrm{negl}(k)$$

for any PPT adversary A. Clearly, every prime-valued collision-resistant function is a division-intractable function family by itself. Division-intractable function families can be more efficient than prime-valued injective functions, see [15] for some constructions.

One can instantiate the root accumulator with a division-intractable function family, by letting the hash function $H \leftarrow \mathcal{H}$ to be a part of the public key pk. One also has to modify the definition of the non-membership proof. Namely, if $m \notin S$, Proof works as follows:

- Let $\delta \leftarrow \prod_{s \in S} H(s)$. For $\gamma \leftarrow \gcd(H(m), \delta)$, find $\alpha, \beta \in R_i$, such that $\alpha \cdot H(m) + \beta \cdot \delta = \gamma$. Let $\mathsf{Proof}_{\mathsf{pk}}(m, S) := (\alpha \circ g, \beta, \gamma)$.

Analogously, verification of non-membership is modified as follows:

- If $p = (q, \beta, \gamma) \in D_i \times R_i \times R_i$, then check whether $H(m) \circ q + \beta \circ d = \gamma \circ g$, $\gamma \mid H(m)$, and $\gamma \neq H(m)$. If it is, then return $\mathsf{NotMember}$, else return Error.

(Note that $\gamma \neq H(m)$ because $\mathcal{H}$ is division-intractable.)

Theorem 3 (Sufficient Conditions (with Division-Intractable Function Families).). *Let $\mathcal{H}$ be a division-intractable function family. Let $\mathcal{R}_{\mathcal{D}}$ be a family of modules over Euclidean rings. (1) If $\mathcal{R}_{\mathcal{D}}$ is a strong divisible root module family, then the root accumulator is collision-resistant. (2) If $\mathcal{R}_{\mathcal{D}}$ is a strong divisible root module family, then the root accumulator is undeniable.*

Proof. (1) Construct a machine B that breaks either the strong divisible root assumption of R_D or the division-intractability of $\mathcal{H}$ using as an oracle an adversary A who can break the collision-resistancy of the root accumulator.

1. B obtains $i \leftarrow \mathsf{Setup}(1^k, \omega)$, and random ω.
2. B obtains his challenge $x \leftarrow D_i$, and $H \leftarrow \mathcal{H}$.
3. B sets $\mathsf{pk} \leftarrow (i, x, H)$.
4. B queries $(m, S, p) \leftarrow A(\mathsf{pk}, \omega)$.
5. B sets $\delta \leftarrow \prod_{s \in S} H(s)$.
6. B sets $\beta^* \leftarrow \gcd(H(m), \delta)$, $\alpha^* \leftarrow H(m)/\beta^*$. If $\alpha^* = 1$, then B aborts.
7. By using the Extended Euclidean Algorithm, B computes γ and γ', such that $\gamma \cdot H(m) + \gamma' \cdot \delta = \beta^*$.
8. B returns $(\gamma \circ x + \gamma' \circ p, \alpha^*, \beta^*)$.

Clearly,

$$\begin{aligned}(\alpha^* \cdot \beta^*) \circ (\gamma \circ x + \gamma' \circ p) =& H(m) \circ (\gamma \circ x + \gamma' \circ p) \\ =& (\gamma \cdot H(m)) \circ x + (\gamma' \cdot H(m)) \circ p \\ =& (\gamma \cdot H(m)) \circ x + (\gamma' \cdot \delta) \circ x = \beta^* \circ x \ .\end{aligned}$$

Thus, if A is successful and B does not abort, then B is successful. But if B aborts, then $\beta^* = H(m)$ and thus B has broken the division-intractability of $\mathcal{H}$.

(2) Construct a machine B that breaks either the strong divisible root assumption of R_D or the division-intractability of $\mathcal{H}$ using as oracle an adversary A who can break the undeniability of the root accumulator.

1. B obtains $i \leftarrow \mathsf{Setup}(1^k, \omega)$, and random ω.
2. B obtains his challenge $x \leftarrow D_i$, and $H \leftarrow \mathcal{H}$.
3. B sets $\mathsf{pk} \leftarrow (i, x, H)$.
4. B queries $(m, d, p, \overline{p}) \leftarrow A(\mathsf{pk}, \omega)$, where $\overline{p} = (q, \beta, \gamma)$.
5. B sets $\alpha^* \leftarrow H(m)/\gamma$. If $\alpha^* = 1$, then B aborts.
6. B returns $(q + \beta \circ p, \alpha^*, \gamma)$.

If A is successful, then $H(m) \circ p = d$ and $H(m) \circ q + \beta \circ d = \gamma \circ g$ for $\gamma \mid H(m)$ and $\gamma \neq H(m)$. Thus,

$$(\alpha^* \gamma) \circ (q + \beta \circ p) = H(m) \circ q + (\beta H(m)) \circ p = H(m) \circ q + \beta \circ d = \gamma \circ g \ .$$

Thus, if A is successful and B does not abort, then B is successful in breaking the strong divisible root assumption. But if B aborts, then B has broken the division-intractability of $\mathcal{H}$. □

Now, we show that independently of the properties of R_D, the family $\mathcal{H}$ must be division-intractable.

Lemma 1. *If the root accumulator is collision-resistant, then $\mathcal{H}$ is division-intractable.*

Proof (Sketch.). By contradiction: assume an adversary finds a pair (m, S), $m \notin S$, such that $\prod_{s \in S} H(s) = \alpha \cdot H(m)$, for some $\alpha \in R_i$. Now,

$$d = (\prod_{s \in S)} H(s) \circ g = (\alpha \cdot H(m)) \circ g = H(m) \circ (\alpha \circ g) = H(m) \circ p$$

with $p = \alpha \circ g$, and therefore the adversary has broken the accumulator. □

6 Relations between New Assumptions

Clearly, if $\mathcal{R}_{\mathcal{D}}$ is a strong root module family, then it is also a strong prime root module family. (If it is difficult to find (y, α) such that $\alpha \circ y = x$, then it is also difficult to find (y, α) with α irreducible, such that $\alpha \circ y = x$.) The

opposite holds only if factorization is easy in the Euclidean ring. Moreover, if $\mathcal{R}_{\mathcal{D}}$ is a strong prime root module family, then it is clearly a root module family. Thus, the strong prime root assumption is in its strength somewhere between the root assumption and the strong root assumption. Because we showed that root accumulator with prime-valued injective function H is secure if and only if the underlying module module is strong root module family, we get

Theorem 4. *Let H be a prime-valued injective function. If factorization is difficult in the underlying Euclidean ring, then the security of root accumulator is based on a security assumption that is weaker than the strong root assumption.*

(In particular, the security of RSA accumulator can be based on an assumption that is weaker than the strong RSA assumption.)

Clearly, if $\mathcal{R}_{\mathcal{D}}$ is a strong divisible root family, then $\mathcal{R}_{\mathcal{D}}$ is also a strong root module family. (To break the strong divisible root assumption, just return $(g, \alpha, 1)$, where (g, α) was returned by an adversary who breaks the strong root assumption.) To show that the proposed strong divisible root assumption in this special case is not too strong, we reduce its security to the strong root assumption conditionally to the small root assumption that generalizes an earlier assumption of the same name by Damgård and Fujisaki [13]. (In their paper, it was assumed that $\beta = 2$.) See [13] for discussion.

Theorem 5. *Let $\mathcal{R}_{\mathcal{D}}$ be defined as always. Let the next two assumptions hold: (a) $\mathcal{R}_{\mathcal{D}}$ is a strong root module family, (b) For any $i \in I$, it is intractable to find elements $g \in D_i$ such that $\alpha \circ g = 0$ for some α with non-minimal non-zero degree* $\deg(\alpha)$*, but $\beta \circ g \neq 0$ for some* $\deg(\beta) < \deg(\alpha)$ *(we call this a* small root assumption*). Then $\mathcal{R}_{\mathcal{D}}$ is a strong divisible root module family.*

Proof. Assume that adversary A breaks the strong divisible root assumption. Construct a machine B that breaks one of the two premises as follows.

1. B obtains $i \leftarrow \mathsf{Setup}(1^k, \omega)$ and ω.
2. B gets his challenge $x \leftarrow D$ of the strong root problem game.
3. B obtains $(y, \alpha, \beta) \leftarrow A(x, \omega)$.
4. B returns (y, α).

Assume that A is successful. Then $(\alpha \cdot \beta) \circ y = \beta \circ x$ and $\alpha \neq 1$. If $\beta = 1$ then we are done. Otherwise, denote $w \leftarrow \alpha \circ y - x$. If $w = 0$ then we are done. Otherwise,

$$\beta \circ w = \beta \circ (\alpha \circ y) - \beta \circ x = (\alpha \cdot \beta) \circ y - \beta \circ x = \beta \circ x - \beta \circ x = 0 \ .$$

Choose some $\beta' \in R$ with $\deg(\beta') < \deg(\beta)$. By the small root assumption, also $\beta' \circ w = 0$. Thus, by application of the Euclidean algorithm, $\alpha = x \cdot \beta' + q$ such that $\deg(q) < \deg(\beta')$. By the choice of β',

$$\alpha \circ w = x \circ (\beta' \circ w) + q \circ w = q \circ w \ .$$

By the small root assumption, $w = 0$ and thus $\alpha \circ y = x$. □

7 Example Instantiations

7.1 RSA Accumulator

In the RSA accumulator, as modified by [1,26], the public parameters contain $n = PQ$ that is a product of two safe primes, and $D_n = \mathbb{Z}_n^*$, R_n with $\alpha \circ g := g^\alpha \pmod n$. If the factorization of n is known to a collusion of parties (say, generated by a malicious server or in a threshold manner by several parties who are all later corrupted), they can jointly compute membership proofs of any element m by defining $p \leftarrow \mathsf{Dig}_{n,g}(S)^{H(m)^{-1} \pmod{\phi(n)}} \pmod n$. Therefore, the RSA accumulator is not collision-resistant without trusted setup.

7.2 Root Accumulator in Class Groups of IQ Order

Class Group Preliminaries. Let Δ be a negative integer such that $\Delta \equiv 0, 1 \pmod 4$. The ring $\mathcal{O}_\Delta = \mathbb{Z} + \frac{\Delta+\sqrt{\Delta}}{2} \cdot \mathbb{Z}$ is an *imaginary quadratic order* of *discriminant* Δ. Its field of fractions is $\mathbb{Q}(\sqrt{\Delta})$. The discriminant Δ is *fundamental* if Δ is square-free if $\Delta \equiv 1 \pmod 4$ or $\Delta/4$ is square-free if $\Delta \equiv 0 \pmod 4$. The ring $\mathcal{O}_\Delta$ is a *maximal order* if Δ is fundamental. The *fractional ideals* of $\mathcal{O}_\Delta$ are of form $q(a\mathbb{Z} + (b + \sqrt{\Delta})/2\mathbb{Z})$ with $q \in \mathbb{Q}$, $a \in \mathbb{Z}^+$, $b \in \mathbb{Z}$ and $4a \mid (b^2 - \Delta)$. Therefore, a fractional ideal can be represented by a triple (q, a, b). An ideal (q, a, b) is *integral* if $q = 1$; an integral ideal can be represented by a pair (a, b). Two fractional ideals $\mathfrak{a}, \mathfrak{b} \subseteq \mathcal{O}_\Delta$ are *equivalent* if for some nonzero $\alpha \in \mathbb{Q}(\sqrt{\Delta})$, $\mathfrak{a} = \alpha\mathfrak{b}$. The set of equivalence classes forms an Abelian group under ideal multiplication; this group is called the *class group* and denoted by $\mathrm{Cl}(\Delta)$. The class group is always finite, its order is called the *class number* and denoted by $h(\Delta) := |\mathrm{Cl}(\Delta)|$.

For an integral ideal there exists a $c \in \mathbb{Z}^+$, such that $\Delta = b^2 - 4ac$. An ideal is called *reduced* if (a) $\gcd(a, b, c) = 1$, (b) $-a < b \leq a \leq c$ and (c) $b \geq 0$ if $a = c$. Every equivalence class contains exactly one reduced ideal. Thus, every element of $\mathrm{Cl}(\Delta)$ can be represented by a reduced ideal of $\mathcal{O}_\Delta$, and checking equality of two ideal classes means comparing the representatives. The neutral element of $\mathrm{Cl}(\Delta)$ is represented by $(1, \Delta \pmod 2)$. The inverse of the ideal class represented by (a, b) is the ideal class represented by $(a, -b)$. The group operation in $\mathrm{Cl}(\Delta)$ is ideal multiplication followed by reduction; a group operation requires $O(\log^2 |\Delta|)$ bit-operations. If (a, b) is reduced then $a \leq \sqrt{|\Delta|/3}$. For more information on computations in class groups see [12, Chapter 5] or [3]; for algorithms see [21,3].

Class Groups and Cryptography. Class groups were first proposed for use in cryptography by Buchmann and Williams [4]. The class number is not efficiently computable if Δ is fundamental, but the even part of $h(\Delta)$ can be efficiently computed if the prime factorization of Δ is known. All problems of Def. 4 can be instantiated to the class groups, and the known efficient algorithms for tackling the discrete logarithm, order, and root problem are tightly connected [3].

General number fields sieve [22], the best currently known factorization algorithm, runs in time $L_n[1/3, \sqrt[3]{\frac{64}{9}}]$. The best currently known algorithm (MPQS)

for the root problem in maximal orders runs in time $L_\Delta[1/2, 1 + o(1)]$ [21]. Even this time is only empirically suggested, the best rigorous algorithm for computing the discrete logarithm runs in time $L_\Delta[\frac{1}{2}, \frac{3}{4}\sqrt{2} + o(1)]$ assuming the Extended Riemann Hypothesis [27]. On the other hand, if $\mathcal{O}_\Delta$ is non-maximal then the discrete logarithm problem in $\mathrm{Cl}(\Delta)$ can be reduced to the discrete logarithm problem in multiplicative groups of finite fields [20]. On the other hand, a $(p-1)$-like algorithm can compute the class number efficiently, given that $h(\Delta)$ is smooth [19]. Hamdy and Möller estimate the probability that a $h(\Delta)$ is B-smooth for randomly chosen k-bit Δ, and conclude that if k-bit discriminants are large enough to guarantee security against the MPQS algorithm, then the probability to find a sufficiently smooth $h(\Delta)$ by choosing k-bit Δ's randomly and applying the $(p-1)$-like algorithm to them, is negligible. Their heuristic, that we also follow, is that the same holds true even when $|\Delta|$ is chosen to be a k-bit random prime.

In general, we rely on the next properties of the class groups:

- If $-\Delta$ is a random k-bit prime, for large k, then computing the roots in the class group $\mathrm{Cl}(\Delta)$ as well as the order of a random element from $\mathrm{Cl}(\Delta)$ is assumed to be intractable. In particular, the length of the discriminant Δ is reasonable: to achieve the same security as with $k = 1536$ in the RSA case, it seems to be sufficient to take $k \approx 1000$ [19].
- if $-\Delta$ is prime and Δ is fundamental, then with high probability 0.9775 the class group $\mathrm{Cl}(\Delta)$ of imaginary quadratic order is cyclic. Moreover, then $h(\Delta)$ is odd.

Root Accumulator in Class Groups of Imaginary Quadratic Order. Let us now concentrate on the case where $R_i = \mathbb{Z}$, and D_i is a class group of imaginary quadratic order, with $\alpha \circ x := x^\alpha$ in D_i.

In the setup phase, for class groups, choose a random negative k-bit prime fundamental discriminant Δ, that is, let $-\Delta$ be a random k-bit prime with $\Delta \equiv 1 \mod 4$. For a k-bit prime i, let $D_i := \mathrm{Cl}(-i)$. Note that for a random k-bit negative Δ, $h(\Delta)$ is not smooth [19], and it was also conjectured in [19] that this also holds when Δ is a random negative k-bit prime.

Another assumption that we make (but that is common to all previous papers on class group-based cryptography) is that with probability $1-\mathrm{negl}(k)$, a random element of $\mathrm{Cl}(\Delta)$ is a generator of some sufficiently large subgroup of $\mathrm{Cl}(\Delta)$.

Word of Caution. While the (weaker) root assumption is a well known assumption in class groups, we are not aware of any a priori use of the strong root assumption in class groups except [13]. Since also [13] did not analyze the strong root assumption but only used it, we must warn that this assumption is yet almost unstudied in the class groups. However, we hope that the current paper provides new incentive to study strong root assumption (and related assumptions) in class groups. A disproof of such assumptions would constitute a major result by itself.

Acknowledgments. The author was supported by Estonian Science Foundation, grant #9303 and the European Union through the European Regional Development Fund. We thank Valdis Laan, Safuat Hamdy and Lauri Tarkkala for useful comments.

References

1. Barić, N., Pfitzmann, B.: Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 480–494. Springer, Heidelberg (1997)
2. Benaloh, J.C., de Mare, M.: One-Way Accumulators: A Decentralized Alternative to Digital Signatures. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 274–285. Springer, Heidelberg (1994)
3. Buchmann, J., Hamdy, S.: A Survey on IQ Cryptography. Technical Report TI-4/01, TU Darmstadt, Fachbereich Informatik (March 21, 2001)
4. Buchmann, J.A., Williams, H.C.: A Key-exchange System Based on Imaginary Quadratic Fields. Journal of Cryptology 1(2), 107–118 (1988)
5. Buldas, A., Laud, P., Lipmaa, H.: Accountable Certificate Management Using Undeniable Attestations. In: Jajodia, S., Samarati, P. (eds.) ACM CCS 2000, Athens, Greece, November 2-4, pp. 9–18. ACM Press (2000)
6. Buldas, A., Laud, P., Lipmaa, H.: Eliminating Counterevidence with Applications to Accountable Certificate Management. Journal of Computer Security 10(3), 273–296 (2002)
7. Buldas, A., Laud, P., Lipmaa, H., Villemson, J.: Time-Stamping with Binary Linking Schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 486–501. Springer, Heidelberg (1998)
8. Buldas, A., Lipmaa, H., Schoenmakers, B.: Optimally Efficient Accountable Time-Stamping. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 293–305. Springer, Heidelberg (2000)
9. Camacho, P., Hevia, A., Kiwi, M., Opazo, R.: Strong Accumulators from Collision-Resistant Hashing. In: Wu, T.-C., Lei, C.-L., Rijmen, V., Lee, D.-T. (eds.) ISC 2008. LNCS, vol. 5222, pp. 471–486. Springer, Heidelberg (2008)
10. Camenisch, J., Kohlweiss, M., Soriente, C.: An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 481–500. Springer, Heidelberg (2009)
11. Camenisch, J., Lysyanskaya, A.: Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002)
12. Cohen, H.: A Course in Computational Algebraic Number Theory. Graduate Texts in Mathematics. Springer (1995)
13. Damgård, I., Fujisaki, E.: A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 125–142. Springer, Heidelberg (2002)
14. Damgård, I., Koprowski, M.: Generic Lower Bounds for Root Extraction and Signature Schemes in General Groups. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 256–271. Springer, Heidelberg (2002)
15. Gennaro, R., Halevi, S., Rabin, T.: Secure Hash-and-Sign Signatures without the Random Oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999)

16. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
17. Haber, S., Stornetta, W.S.: How to Time-Stamp a Digital Document. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 437–455. Springer, Heidelberg (1991)
18. Hamdy, S.: Computations in Class Groups of Imaginary Quadratic Number Fields. In: Innovations in Information Technology, Dubai, UAE, November 19-21, pp. 1–5 (2006)
19. Hamdy, S., Möller, B.: Security of Cryptosystems Based on Class Groups of Imaginary Quadratic Orders. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 234–247. Springer, Heidelberg (2000)
20. Hühnlein, D., Takagi, T.: Reducing Logarithms in Totally Non-maximal Imaginary Quadratic Orders to Logarithms in Finite Fields. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 219–231. Springer, Heidelberg (1999)
21. Jacobson Jr., M.J.: Subexponential Class Group Computation in Quadratic Orders. PhD thesis, Technische Universität Darmstadt, Fachbereich Informatik, Darmstadt, Germany (1999)
22. Lenstra, A.K., Lenstra, J. H.W. (eds.): The Development of the Number Field Sieve. Lecture Notes in Mathematics, vol. 1554. Springer, Heidelberg (1993)
23. Li, J., Li, N., Xue, R.: Universal Accumulators with Efficient Nonmembership Proofs. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 253–269. Springer, Heidelberg (2007)
24. Nguyen, L.: Accumulators from Bilinear Pairings and Applications. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 275–292. Springer, Heidelberg (2005)
25. Sander, T.: Efficient Accumulators without Trapdoor Extended Abstract. In: Varadharajan, V., Mu, Y. (eds.) ICICS 1999. LNCS, vol. 1726, pp. 252–262. Springer, Heidelberg (1999)
26. Sander, T., Ta-Shma, A., Yung, M.: Blind, Auditable Membership Proofs. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 53–71. Springer, Heidelberg (2001)
27. Vollmer, U.: Asymptotically Fast Discrete Logarithms in Quadratic Number Fields. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 581–594. Springer, Heidelberg (2000)