# Compact Accumulator using Lattices

Mahabir Prasad Jhanwar
R. C. Bose Center for Cryptology and Security
Indian Statistical Institute, India

Reihaneh Safavi-Naini
Department of Computer Science
University of Calgary, Canada

**Abstract**

An accumulator is a *succinct* aggregate of a set of values where it is possible to issue *short* membership proofs for each accumulated value. A party in possession of such a membership proof can then demonstrate that the value is included in the set. In this paper, we preset the first lattice-based accumulator scheme that issues compact membership proofs. The security of our scheme is based on the hardness of Short Integer Solution ($\mathsf{SIS}$) problem.

## 1 Introduction

**Accumulators:** An accumulator scheme ($\mathcal{AC}$) is a cryptographic authentication primitive for *optimally* verifying set-membership relations. Briefly, given a set $S$ of elements, an $\mathcal{AC}$ scheme can compute a short representation $\mathsf{Acc}(S)$ of $S$, called *accumulation value*, such that for every element $x \in S$ a short membership *witness* $w_x$ of "$x$ belonging to $S$" can be generated. The accumulation value $\mathsf{Acc}(S)$ is published, and everybody can obtain it in an authenticated manner. Later, by exhibiting a *valid* $(x, w_x)$ pair, a prover can convince a verifier that the value $x$ was indeed accumulated into $\mathsf{Acc}(S)$. The security of the scheme requires that it be difficult to find a valid value-witness pair $(x^*, w_{x^*})$ such that $x^* \notin S$. An accumulator is *compact* if it yields accumulation values and witnesses that are of constant size (i.e., independent of the number of elements $S$ contains).

**Applications:** Accumulators have proven to be a very strong mathematical tool with applications in a variety of privacy preserving technologies. Applications of accumulators include efficient time-stamping [BdM93], anonymous credential systems and group signatures [Nyb96, Ngu05, CKS09], ring signatures [DKNS04], redactable signatures [PS14], sanitizable signatures [CJ10], P-homomorphic signatures [ABC$^+$12], and Zerocoin [MGGR] (an extension of the cryptographic currency Bitcoin), etc.

**Evaluation:** Accumulators were first introduced by Benaloh and de Mare [BdM93], and were later further studied and extended by Baric and Pfitzmann [BP97]. The security of both constructions was proved under the *strong* RSA assumption. Camenisch and Lysyanskaya [CL02] augmented the latter work and proposed *dynamic* accumulators, in which elements can be efficiently added to and removed from the set of accumulated values, as well as privacy-preserving membership proofs. Alternative constructions of dynamic accumulators based on bilinear pairing [Ngu05, DT08, CKS09], Paillier's trapdoor permutation [WWP08], and vector commitments [CF13] are also known. Li et al. [LLX] introduced *universal* accumulators that extend the functionality of accumulators by supporting proofs that a given element is not a member of the set that has been accumulated. The security of their proposed instantiation is based on strong RSA assumption. Camacho et al. [CHKO12] and Buldas et al. [BLL00] independently introduced *strong* universal accumulators (also known as *undeniable* accumulators), which do not assume the accumulator manager is trusted. Both constructions were proved secure under the assumption that collision-resistant hash functions exists.

### 1.1 Our Contribution

In recent years, there has been rapid development in the use of lattices for constructing rich cryptographic schemes (these include digital signatures [GPV08, Boy10, CHKP12], identity-based encryption [GPV08] and hierarchical IBE [CHKP12, ABB10], non-interactive zero knowledge [PV08], and even a fully homomorphic cryptosystem [Gen09]). Among other reasons, this is because such schemes have yet to be broken by quantum algorithms, and their security can be based solely on *worst-case* computational assumptions.

In the spirit of lattice-based cryptography, we present the first compact accumulator scheme from lattices and prove that it is secure based on the hardness of Short Integer Solution ($\mathsf{SIS}$) problem. As the average-case $\mathsf{SIS}$ problem was shown to be as hard as certain worst-case lattice problems [Ajt96, MR07, GPV08], our scheme owns provable security under worst-case hardness assumption.

### 1.2 Related Work

Although, there exists no direct lattice-based $\mathcal{AC}$ scheme, the constructions in [BdM93, BLL00, CHKO12] give indirect lattice-based instantiations because they only assume collision-resistant hash functions exist. This is true as lattice-based constructions of collision-resistant hash function are known [LM06, PR06], and therefore the security of the resulting schemes can also be reduced to worst-case assumptions on lattices. However, hash-tree based $\mathcal{AC}$ schemes are not *compact* as, in this setting, witnesses grows logarithmically with the number of elements in $S$.

## 2 Preliminaries

**Notation:** Let $\lambda \in \mathbb{N}$ be the security parameter and $1^\lambda$ its unary representation. We use standard asymptotic notation to describe the order of growth of functions. For any positive real valued functions $f(n)$ and $g(n)$ we write $f = O(g)$ if there exists two constants $c_1, c_2$ such that $f(n) < c_1 \cdot g(n)$ for all $n \geq c_2$; $f = \Omega(g)$ if $g = O(f)$; $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$; and $f = o(g)$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. We denote $f = \tilde{O}(g)$ if $f = O(g \cdot \mathsf{poly}(\log g))$. The notation $\tilde{\Theta}$ is defined analogously. We denote $\omega(f(n))$ to denote a function that grows faster than $c \cdot f(n)$ for any $c > 0$. We let $\mathsf{poly}(n)$ denote an unspecified function $f(n) = O(n^c)$ for some constant $c$. A function $f(n)$ is called negligible, often written as $f(n) = \mathsf{negl}(n)$, if $f = o(\frac{1}{g})$ for any polynomial $g = \mathsf{poly}(n)$. A function of $n$ is called *overwhelming* if it is $1 - \mathsf{negl}(n)$. For a positive integer $k$, let $[k]$ denote the set $\{1, \ldots, k\}$. We denote the set of integers modulo $q$ by $\mathbb{Z}_q$, and identify it with the set $\{0, \ldots, q-1\}$ in the natural way. Column vectors are name by lower-case bold letters (e.g., $\boldsymbol{b}$) and matrices by upper-case bold letters (e.g., $\boldsymbol{B}$). For a matrix $\boldsymbol{S} \in \mathbb{R}^{m_1 \times m_2}$, we call the norm of $\boldsymbol{S}$ as $||\boldsymbol{S}|| = \max_{1 \leq i \leq m_2} ||\boldsymbol{s}_i||$, where $||\boldsymbol{s}_i||$ denotes the $\ell_2$-norm (Euclidean norm) of the column vector $\boldsymbol{s}_i$. We let $\tilde{\boldsymbol{S}} \in \mathbb{R}^{m_1 \times m_2}$ denotes the matrix whose columns $\tilde{\boldsymbol{s}}_1, \ldots, \tilde{\boldsymbol{s}}_{m_2}$ represents the Gram-Schmidt orthogonalization of the vectors $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{m_2}$ taken in the same order. Let $||\tilde{\boldsymbol{S}}||$ denote the Gram-Schmidt norm of $\boldsymbol{S}$.

### 2.1 Lattices

Let $\mathbb{R}^m$ be the $m$-dimensional Euclidean space. A *lattice* in $\mathbb{R}^m$ is the set

$$\mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k) = \left\{ \sum_{i=1}^{k} c_i \boldsymbol{b}_i \mid c_i \in \mathbb{Z} \right\}. \tag{1}$$

of all integral combination of $k$ linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ in $\mathbb{R}^m$ $(m \geq k)$ [1]. The integers $k$ and $m$ are called the *rank* and *dimension* of the lattice, respectively. The sequence of vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ is called a *lattice basis* and it is conveniently represented as a matrix $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k] \in \mathbb{R}^{m \times k}$ having the basis vectors as columns. Using the matrix notation, (1) can be written in a more compact form as $\mathcal{L}(\boldsymbol{B}) = \{\boldsymbol{Bc} \mid \boldsymbol{c} \in \mathbb{Z}^k\}$, where $\boldsymbol{Bc}$ is the usual matrix-vector multiplication. When $m = k$, the lattice is called *full-rank*. A lattice $\Lambda$ is called *integer lattice* if $\Lambda \subseteq \mathbb{Z}^m$. In this work, every lattice will be a full-rank lattice.

The *minimum distance* $\lambda_1(\Lambda)$ of a lattice $\Lambda$ is the length (Euclidean length, i.e., $\ell_2$ norm, unless otherwise indicated). More generally, the $i$th *successive minimum* $\lambda_i(\Lambda)$ is the smallest radius $r$ such that $\Lambda$ contains $i$ linearly independent vectors of norm at most $r$. The following are the two standard worst-case approximation problems on lattices: Shortest Vector Problem ($\mathsf{SVP}_\gamma$) and Shortest Independent Vector Problem ($\mathsf{SIVP}_\gamma$) . In both problems, $\gamma = \gamma(m)$ is the approximation factor as a function of the lattice-dimension.

**Definition 1** ($\mathsf{SVP}_\gamma$) *An input to* $\mathsf{SVP}_\gamma$ *is a basis* $\boldsymbol{B}$ *of a full-rank* $m$*-dimensional lattice. The goal is to output a nonzero lattice vector* $\boldsymbol{Bx}$ *(with* $\boldsymbol{x} \in \mathbb{Z}^m \backslash \{\boldsymbol{0}\}$*) such that* $||\boldsymbol{Bx}|| \leq \gamma \cdot ||\boldsymbol{By}||$ *for any* $\boldsymbol{y} \in \mathbb{Z}^m \backslash \{\boldsymbol{0}\}$.

**Definition 2** ($\mathsf{SIVP}_\gamma$) *An input to* $\mathsf{SIVP}_\gamma$ *is a basis* $\boldsymbol{B}$ *of a full-rank* $m$*-dimensional lattice. The goal is to output a set of* $m$ *linearly independent lattice vectors* $\boldsymbol{Bx}_1, \ldots, \boldsymbol{Bx}_m \in \mathcal{L}(\boldsymbol{B})$ *such that* $max_i\{||\boldsymbol{Bx}_i||\} \leq \gamma \cdot \lambda_m(\mathcal{L}(\boldsymbol{B}))$.

#### 2.1.1 $q$-ary Lattices

In this work we use $q$-ary lattices; a special family of full-rank integer lattices. A lattice from this family is most naturally specified not by a basis, but instead by a parity check matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ for some positive integer $n$ and positive integer modulus $q$. The associated full rank lattice of dimensional $m$ is defined as:

$$\Lambda^{\perp}(\boldsymbol{A}) \quad = \quad \{\boldsymbol{x} \in \mathbb{Z}^m \mid \boldsymbol{Ax} = \boldsymbol{0} \bmod q\} \tag{2}$$

It is routine to check that $\Lambda^{\perp}(\boldsymbol{A})$ contains $\boldsymbol{0} \in \mathbb{Z}^m$ (thus non-empty) and is closed under subtraction, hence it is a lattice. The hardness of these lattices is most naturally parametrized by $n$ (not $m$, even though $m$ is the dimension of the lattices) and therefore it is standard to consider the parameters $m = m(n)$ and $q = q(n)$ as functions of $n$. By taking $m = c \cdot n \log q$ for some constant $c \geq 1$, it can be shown that with high probability, the minimum distance $\lambda_1\left(\Lambda^{\perp}(\boldsymbol{A})\right)$ of $\Lambda^{\perp}(\boldsymbol{A})$ is at most $\Theta(\sqrt{n \log q})$, where $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ is random.

Ajtai [Ajt99], Alwen and Peikert [AP09], Micciancio and Peikert [MP12] provided methods to produce a matrix $\boldsymbol{A}$ statistically close to uniform in $\mathbb{Z}_q^{n \times m}$ along with a short basis $\boldsymbol{T_A}$ of lattice $\Lambda^{\perp}(\boldsymbol{A})$. It is summarized in the following lemma.

[1] Alternatively, lattices can also be characterized without any reference to any basis. A lattice $\Lambda$ can be defined as a discrete nonempty subset of $\mathbb{R}^m$ which is closed under subtraction, i.e., if $\boldsymbol{x} \in \Lambda$ and $\boldsymbol{y} \in \Lambda$, then also $\boldsymbol{x} - \boldsymbol{y} \in \Lambda$. Here *discrete* means that there exists a positive real $\lambda > 0$ such that the Euclidean distance between any two lattice vectors is at least $\lambda$.

**Proposition 1 (Short Basis Generation)** *There is a PPT algorithm that, on input a security parameter $1^\lambda$, an odd prime $q = \mathsf{poly}(\lambda)$, and two integers $n = \Theta(\lambda)$ and $m \geq 6n\log q$, outputs a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ statistically close to uniform, and a basis $\boldsymbol{T_A}$ for $\Lambda^\perp(\boldsymbol{A})$ with overwhelming probability such that $||\tilde{\boldsymbol{T}}_{\boldsymbol{A}}|| \leq \tilde{\Theta}(\sqrt{m})$.*

We refer to the algorithm of Proposition 1 by $\mathsf{TrapGen}(1^\lambda)$.

**Primitive Matrix:** We say that a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ is *primitive* if its columns generate all of $\mathbb{Z}_q^n$, i.e., $\boldsymbol{A}\cdot\mathbb{Z}^m \pmod q = \mathbb{Z}_q^n$. It is known that for any fixed constant $C > 1$ and any $m \geq Cn\log q$, a uniformly random $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ is primitive, except with $2^{-\Omega}(n) = \mathsf{negl}(n)$ probability. Therefore, throughout the paper we implicitly assume that such a uniform $\boldsymbol{A}$ is primitive.

#### 2.1.2 Hardness Assumption

The *short integer solution* (SIS) problem was first suggested to be hard on average by Ajtai [Ajt96] and later in [MR07] was formalized as follows. The security of our accumulator scheme is based on the hardness of this problem.

**Definition 3 (SIS Problem)** *The small integer solution problem* SIS *(in the $\ell_2$ norm) is as follows: given an integer $q$, a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, and a real $\beta$, find an integer vector $\boldsymbol{e} \in \mathbb{Z}^m$ such that $\boldsymbol{Ae} = \boldsymbol{0} \bmod q$ and $||\boldsymbol{e}|| \leq \beta$.*

Clearly, the problem is syntactically equivalent to finding some short nonzero vector in $\Lambda^\perp(\boldsymbol{A})$. For functions $q(n), m(n)$, and $\beta(n)$, an *average-case* SIS problem instance is drawn from the probability ensemble over instances $(q(n), \boldsymbol{A}, \beta(n))$ where $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ is uniformly random. This average-case problem was shown to be as hard as certain worst-case lattice problems, first by Ajtai [Ajt96], then by Micciancio and Regev [MR07], and Gentry et al. [GPV08].

**Theorem 1** *( [GPV08]) For any poly-bounded $m$, any $\beta = \mathsf{poly}(n)$ and for any prime $q \geq \beta\cdot\omega(\sqrt{n\log n})$, the average-case $\mathsf{SIS}_{q,m,\beta}$ is as hard as approximating the Shortest Independent Vector Problem ($\mathsf{SIVP}_\gamma$), among others, in the worst-case to within certain $\gamma = \beta\cdot\tilde{O}(\sqrt{n})$ factors.*

#### 2.1.3 Discrete Gaussian Distribution over Lattices

For any $s > 0$ the Gaussian function $\rho_{s,\boldsymbol{c}} : \mathbb{R}^n \to \mathbb{R}$ centered at $\boldsymbol{c} \in \mathbb{R}^n$ with parameter $s$ is defined as:

$$\forall \boldsymbol{x} \in \mathbb{R}^n,\ \rho_{s,\boldsymbol{c}}(\boldsymbol{x}) = e^{-\frac{\pi||\boldsymbol{x}-\boldsymbol{c}||^2}{s^2}}.$$

For any $\boldsymbol{c} \in \mathbb{R}^n$, real $s > 0$, and $n$-dimensional lattice $\Lambda$, define the discrete Gaussian distribution $D_{\Lambda,s,\boldsymbol{c}}$ over $\Lambda$ (with center $\boldsymbol{c}$ and Gaussian parameter $s$) as:

$$\forall \boldsymbol{x} \in \mathbb{R}^n,\ D_{\Lambda,s,\boldsymbol{c}}(\boldsymbol{x}) = \frac{\rho_{s,\boldsymbol{c}}(\boldsymbol{x})}{\rho_{s,\boldsymbol{c}}(\boldsymbol{\Lambda})}.$$

Micciancio and Regev [MR07] proved that the norm ($\ell_2$ norm) of vectors sampled from the discrete Gaussian distribution is small with high probability. We preset this result specialized to $q$-ary lattices.

**Lemma 1** *Let $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ be a primitive matrix, and $s$ be a Gaussian parameter with $s \geq \omega(\sqrt{\log m})$. Then for $m$-dimensional full-rank lattice $\Lambda^\perp(\boldsymbol{A})$, and $\boldsymbol{c} \in \mathbb{R}^m$,*

$$\mathsf{Pr}_{\boldsymbol{x}\leftarrow D_{\Lambda^\perp(\boldsymbol{A}),s,\boldsymbol{c}}}\left[\ ||\boldsymbol{x}-\boldsymbol{c}|| > s\sqrt{m}\ \right] \leq \mathsf{negl}(n).$$

Gentry et al. [GPV08] show that, given a basis $\boldsymbol{B}$ for a lattice $\Lambda$, one can efficiently sample points in $\Lambda$ with discrete Gaussian distribution for sufficiently large values of $s$.

**Theorem 2** *There is a* PPT *algorithm that, given a basis* $\boldsymbol{B}$ *of an* $m$*-dimensional lattice* $\Lambda$*, a parameter* $s \geq ||\tilde{\boldsymbol{B}}|| \cdot \omega(\sqrt{\log m})$*, and a center* $\boldsymbol{c} \in \mathbb{R}^m$*, outputs a sample from a distribution that is statistically close to* $D_{\Lambda,s,\boldsymbol{c}}$*.*

We refer to the algorithm of Theorem 2 by $\mathsf{SampleD}(\boldsymbol{B}, s, \boldsymbol{c})$.

The Gaussian Sampling Algorithm: $\mathsf{SampleD}(\boldsymbol{B}, s, \boldsymbol{c})$

- Input :
  - a lattice $\Lambda \subseteq \mathbb{R}^n$ with a basis $\boldsymbol{B}$,
  - a positive real parameter $s \geq ||\tilde{\boldsymbol{B}}|| \cdot \omega(\sqrt{\log m})$, and
  - a center vector $\boldsymbol{c} \in \mathbb{R}^n$.
- Output :
  - a fresh random lattice vector $\boldsymbol{x} \in \Lambda$ drawn from a distribution statistically close to $D_{\Lambda,s,\boldsymbol{c}}$.

### 2.1.4 Basis Delegation

In [CHKP12] a deterministic polynomial-time algorithm is given to extend a basis of $\Lambda^{\perp}(\boldsymbol{A})$ to a basis (without any loss of quality) of an arbitrary higher-dimensional extension $\Lambda^{\perp}(\boldsymbol{A}||\bar{\boldsymbol{A}})$. We refer to this algorithm by $\mathsf{BasisDel}$.

The Basis Delegation Algorithm: $\mathsf{BasisDel}(\boldsymbol{T_A}, \boldsymbol{A}, \bar{\boldsymbol{A}})$

- Input :
  - an arbitrary $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ such that $\boldsymbol{A}$ is primitive,
  - an arbitrary basis $\boldsymbol{T_A}$ of $\Lambda^{\perp}(\boldsymbol{A})$, and
  - an arbitrary $\bar{\boldsymbol{A}} \in \mathbb{Z}_q^{n\times \bar{m}}$.
- Output :
  - a basis $\boldsymbol{T_{A'}}$ of $\Lambda^{\perp}(\boldsymbol{A'} = \boldsymbol{A}||\bar{\boldsymbol{A}}) \subseteq \mathbb{Z}^{m+\bar{m}}$ such that $||\tilde{\boldsymbol{T}}_{\boldsymbol{A'}}|| = ||\tilde{\boldsymbol{T}}_{\boldsymbol{A}}||$.

### 2.1.5 Cryptographic Accumulators

We now give a formal definition of a cryptographic accumulator scheme.

**Definition 4 (Accumulator Scheme)** *Let* $\mathcal{M}$*,* $\mathcal{C}$ *and* $\mathcal{W}$ *be three sets (the message set, the set containing accumulated values and the set containing witnesses respectively). An accumulator scheme* $\mathcal{AC}$ *is a tuple of* PPT *algorithms (*Setup*,* Accumulate*,* WitGen*,* Verify*) with the following functionalities:*

- $\mathsf{Setup}(1^{\lambda})$*: Given a security parameter* $\lambda$*, it outputs a public key* pk *and a secret key* sk*. The remaining algorithms take* pk *as an implicit input.*
- $\mathsf{Accumulate}(X)$*: If* $X \subseteq \mathcal{M}$ *then it accumulates all the elements of* $X$ *into an accumulation value* $\mathsf{Acc}_X \in \mathcal{C}$*.*
- $\mathsf{WitGen}(X, x, \mathsf{sk})$*: If* $x \in X$ *and* $X \subseteq \mathcal{M}$*, then it outputs a membership witness* $w_x \in \mathcal{W}$*; otherwise it outputs "*$\perp$*" denoting* Error*.*

- $\mathsf{Verify}(x, w_x, c)$: *For $x \in \mathcal{M}$, $w_x \in \mathcal{W}$ and $c \in \mathcal{C}$ it outputs either "1" denoting* member *or "0" denoting* Error*.*

The *correctness* of an accumulator scheme requires that correctly accumulated values have valid witnesses with overwhelming probability, i.e., for $x \in \mathcal{M}$ and $X \subseteq \mathcal{M}$, the verification algorithm $\mathsf{Verify}(x, \mathsf{WitGen}(X, x, \mathsf{sk}), \mathsf{Accumulate}(X))$ outputs 1 if $x \in X$, and 0 otherwise.

**Definition 5 (One-way Security)** *An accumulator scheme is one-way secure* [2] *if, for all polynomial time adversaries $\mathcal{A}$:*

$$\Pr[\mathsf{pk} \leftarrow \mathsf{Setup}(1^\lambda); (X^*, x^*, w_{x^*}) \leftarrow \mathcal{A}(\mathsf{pk}) \mid x^* \notin X^* \subseteq \mathcal{M} \text{ and}$$
$$\mathsf{Verify}(x^*, w_{x^*}, c \leftarrow \mathsf{Accumulate}(X^*)) = 1] \leq \mathsf{negl}(\lambda).$$

If an accumulator satisfies this definition, then it is infeasible for an adversary to prove that a value $x$ was accumulated in a accumulation value $c$ when in fact it was not.

# 3 A Compact Accumulator Scheme

In this section we provide our accumulator scheme from lattices. Then, we look into the proper parameter sizes for our construction in order for the correctness to go through. The security analysis of our scheme will be given in § 3.2. The parameters of our scheme involves:

- a security parameter $1^\lambda$;
- integers $n$ and $q$ (a prime) with $n = \Theta(\lambda)$ and $q = \mathsf{poly}(n)$;
- a dimension $m \geq 6n \lg q$ and a bound $L = m^{1.5}$;
- a Gaussian parameter $s \geq L \cdot \omega\left(\sqrt{\log(m + m')}\right)$, where $m' = \mathsf{poly}(\lambda) \in \mathbb{N}$;
- a message set $\mathcal{M} = \left\{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_r \in \mathbb{Z}_q^{m \times m'} \mid ||\boldsymbol{B}_i|| \leq s\sqrt{m + m'}, 1 \leq i \leq r = \mathsf{poly}(\lambda) \in \mathbb{N}\right\}$[3].

The scheme is defined as follows.

- $\mathsf{Setup}(1^\lambda)$: It uses the algorithm $\mathsf{TrapGen}(1^\lambda)$ from Proposition 1 to generate $(\boldsymbol{A}, \boldsymbol{T_A})$, where $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ is statistically close to uniform and $\boldsymbol{T_A}$ is a short basis of $\Lambda^\perp(\boldsymbol{A})$ with $||\tilde{\boldsymbol{T}}_{\boldsymbol{A}}|| \leq L$. The public key $\mathsf{pk}$ is set to $\boldsymbol{A}$, and the secret key $\mathsf{sk}$ is set to $\boldsymbol{T_A}$. In the following, the other algorithms take $\mathsf{pk} = \boldsymbol{A}$ as an implicit input.

- $\mathsf{Accumulate}(X \subset \mathcal{M})$: Without loss of generality, suppose $X = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_\ell\}$ for some $\ell \in [r]$. It accumulates the $\ell$ matrices in the set $X$ into an accumulator value

$$\mathsf{Acc}_X = \left[\sum_{\boldsymbol{B}_i \in X} \boldsymbol{B}_i\right] \in \mathbb{Z}_q^{m \times m'} \quad .$$

- $\mathsf{WitGen}(X, \boldsymbol{B}, \mathsf{sk})$: Let $X = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_\ell\}$ for some $\ell \in [r]$. If $\boldsymbol{B} \notin X$, return $\perp$. Otherwise, $\boldsymbol{B} \in X$ and let $\boldsymbol{B} = \boldsymbol{B}_j$ for some $j \in [\ell]$. The witness generation algorithms returns a witness $w_{\boldsymbol{B}}$ that $\boldsymbol{B}$ has been accumulated in $\mathsf{Acc}_X$. It first computes the matrix

$$\boldsymbol{F_B} = \left[\boldsymbol{A} || \boldsymbol{A} \cdot \sum_{1 \leq i (\neq j) \leq \ell} \boldsymbol{B}_i\right] \in \mathbb{Z}_q^{n \times (m + m')}.$$

[2] In the literature, the one-way secure accumulators are also known as collision-resistant accumulators.
[3] See Remark 1

It then samples a vector $\boldsymbol{d_B} \in \Lambda^{\perp}(\boldsymbol{F_B}) \subseteq \mathbb{Z}^{(m+m')}$ such that it follows the distribution $D_{\Lambda^{\perp}(\boldsymbol{F_B}),s,\boldsymbol{0}}$. This is done, using $\mathsf{sk} = \boldsymbol{T_A}$, as follows:

$$\boldsymbol{d_B} \leftarrow \mathsf{SampleD}\left(\mathsf{BasisDel}\left(\boldsymbol{T_A}, \boldsymbol{A}, \boldsymbol{A} \cdot \sum_{1 \le i(\neq j) \le \ell} \boldsymbol{B}_i\right), s, \boldsymbol{0}\right).$$

The witness $w_{\boldsymbol{B}}$ is set to $w_{\boldsymbol{B}} = \boldsymbol{d_B}$. See, Theorem 2 for a description of SampleD and § 2.1.4 for BasisDel.

- $\mathsf{Verify}(\boldsymbol{B}, w_{\boldsymbol{B}}, \mathsf{Acc}_X)$: The verification algorithm proceeds as follows:
  - It checks if $||\boldsymbol{B}|| \le s\sqrt{m+m'}$.
  - If yes, it computes

    $$\boldsymbol{F_B} = [\boldsymbol{A} || \boldsymbol{A} \cdot (\mathsf{Acc}_X - \boldsymbol{B})] \in \mathbb{Z}_q^{n \times (m+m')}$$

    and checks if $\boldsymbol{F_B} \cdot w_{\boldsymbol{B}} = 0 \bmod q$, i.e., if $w_{\boldsymbol{B}} \in \Lambda^{\perp}(\boldsymbol{F_B})$.
  - If yes, it finally examine if $w_{\boldsymbol{B}}$ is small by checking $0 < ||w_{\boldsymbol{B}}|| \le s \cdot \sqrt{m+m'}$.

  If all the verifications pass, it outputs 1; otherwise, it outputs 0.

**Remark 1** *In the construction above, we accumulate low-norm matrices $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_r \in \mathbb{Z}_q^{m \times m}$. Depending on the application, one would want to accumulate an arbitrary set $\mathcal{U} = \{u_1, \ldots, u_r\}$ of size $r$. In this case, the issuer of the accumulator would need to publish a mapping from this set to the $\boldsymbol{B}_i$'s that get actually accumulated (under the condition that $max_i\{\log_2 u_i\} \approx max_i\{\log_2 \boldsymbol{B}_i\}$). The following lemma gives a method to obtain low norm matrices.*

***Lemma 2*** *Let $\boldsymbol{B}$ be a $m \times m'$ matrix chosen at random from $\{-1,1\}^{m \times m'}$. Then there is universal constant $C$ such that*

$$\mathsf{Pr}\left[||\boldsymbol{B}|| > C\sqrt{m+m'}\right] < e^{-(m+m')}.$$

*The lemma is given [ABB10] (See Lemma 15). The proof follows from a result of Litvak et al [LPRTj05], where taking $C = 12$ is sufficient. In the setting of our scheme, if we identify $\mathbb{Z}_q$ ($q$ being a prime) with the set $\{-\frac{q+1}{2}, \ldots, -1, 0, 1, \ldots, \frac{q-1}{2}\}$, then a random matrix $\boldsymbol{B} \in \{-1,1\}^{m \times m'}$ is also a matrix in $\mathbb{Z}_q^{m \times m'}$. Lemma 2 further implies that, with high probability, $||\boldsymbol{B}|| \le s\sqrt{m+m'}$ (as, the Gaussian parameter $s > C$).*

### 3.1 Correctness

It is easy to see by inspection that the accumulator scheme is correct, i.e., the correctly accumulated values have verifying witnesses with overwhelming probability. But for completeness, in the following, we discuss the correctness of our scheme.

Let $X = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_\ell\} \subseteq \mathcal{M}$, with corresponding accumulation value $\mathsf{Acc}_X = \sum_{i=1}^{\ell} \boldsymbol{B}_i$. We show that every $\boldsymbol{B} \in X$ admits a verifying witness with respect to $\mathsf{Acc}_X$. Without loss of generality, let $\boldsymbol{B} = \boldsymbol{B}_1$. A valid witness for $\boldsymbol{B}_1$ is a short vector $\boldsymbol{d}_{\boldsymbol{B}_1}$ in the lattice $\Lambda^{\perp}(\boldsymbol{F}_{\boldsymbol{B}_1})$ (where $\boldsymbol{F}_{\boldsymbol{B}_1} = \left[\boldsymbol{A} || \boldsymbol{A} \cdot \sum_{i=2}^{\ell} \boldsymbol{B}_i\right] \in \mathbb{Z}_q^{n \times (m+m')}$), i.e., $||\boldsymbol{d}_{\boldsymbol{B}_1}|| \le s\sqrt{(m+m')}$. The sampling algorithm of Theorem 2 is used to sample such a vector. Lemma 1 says that a sample, following $D_{\Lambda^{\perp}(\boldsymbol{F}_{\boldsymbol{B}_1}),s,\boldsymbol{0}}$, in $\Lambda^{\perp}(\boldsymbol{F}_{\boldsymbol{B}_1})$ has norm bounded by $s\sqrt{(m+m')}$ if $s \ge \omega\left(\sqrt{\log(m+m')}\right)$. The algorithm of Theorem 2 can sample from $D_{\Lambda^\perp(\boldsymbol{F}_{\boldsymbol{B}_1}),s,\boldsymbol{0}}$ if it is provided with a basis $\boldsymbol{T}_{\boldsymbol{F}_{\boldsymbol{B}_1}}$ of $\Lambda^\perp(\boldsymbol{F}_{\boldsymbol{B}_1})$, such that $s \geq ||\tilde{\boldsymbol{T}}_{\boldsymbol{F}_{\boldsymbol{B}_1}}|| \cdot \omega\left(\sqrt{\log(m+m')}\right)$. We now see that this is indeed the case.

The witness generation algorithm has access to a short basis $\boldsymbol{T}_{\boldsymbol{A}}$ of the lattice $\Lambda^\perp(\boldsymbol{A})$. With $\left(\boldsymbol{T}_{\boldsymbol{A}}, \boldsymbol{A}, \boldsymbol{A} \cdot \sum_{i=2}^{\ell} \boldsymbol{B}_i\right)$ as input, the basis delegation algorithm $\mathsf{BasisDel}$ of § 2.1.4 constructs a basis $\boldsymbol{T}_{\boldsymbol{F}_{\boldsymbol{B}_1}}$ of $\Lambda^\perp(\boldsymbol{F}_{\boldsymbol{B}_1})$ such that $||\tilde{\boldsymbol{T}}_{\boldsymbol{F}_{\boldsymbol{B}_1}}|| = ||\tilde{\boldsymbol{T}}_{\boldsymbol{A}}||$. But, as $||\tilde{\boldsymbol{T}}_{\boldsymbol{A}}|| \leq L \leq \frac{s}{\omega\left(\sqrt{\log(m+m')}\right)}$, therefore we have $s \geq ||\tilde{\boldsymbol{T}}_{\boldsymbol{F}_{\boldsymbol{B}_1}}|| \cdot \omega\left(\sqrt{\log(m+m')}\right)$.

Hence, the sampled vector $\boldsymbol{d}_{\boldsymbol{B}_1} \leftarrow \mathsf{SampleD}\left(\mathsf{BasisDel}\left(\boldsymbol{T}_{\boldsymbol{A}}, \boldsymbol{A}, \boldsymbol{A} \cdot \sum_{i=2}^{\ell} \boldsymbol{B}_i\right), s, \boldsymbol{0}\right)$ constitute a valid witness for the membership of $\boldsymbol{B}_1$ in $X$ with respect to $\mathsf{Acc}_X$.

### 3.2 Security

In the following theorem we now reduce the $\mathsf{SIS}$ problem to break the security of our accumulator scheme.

**Theorem 3** *For parameters $\lambda, n, q, m, m', L, s,$ and $r$, as listed in the scheme, if there is a* PPT *adversary $\mathcal{A}$ that breaks the one-way security of our accumulator scheme, then there is a* PPT *algorithm $\mathcal{B}$ that solves the $\mathsf{SIS}_{q,m,\beta}$ problem for some polynomial function $\beta = \mathsf{poly}(\lambda)$; in particular $\beta = rs^3(m+m')$* [4]*.*

**Proof**: Suppose that there exists such a forger $\mathcal{A}$. We construct a solver $\mathcal{B}$ that simulates an attack environment and uses an invalid element-witness pair ($\mathcal{A}$'s output) to create its solution for $\mathsf{SIS}$ problem. The various operations performed by $\mathcal{B}$ are the following.

- **Invocation**
  - $\mathcal{B}$ is invoked on a random instance $(q, \boldsymbol{A} \in \mathbb{Z}_q^{n\times m}, \beta)$ of $\mathsf{SIS}$ problem and asked to submit a solution.
- **Simulation**
  - $\mathcal{B}$ sets the public key $\mathsf{pk}$ of accumulator scheme to $\mathsf{pk} = \boldsymbol{A}$.
  - It then chooses a set $\mathcal{M} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_r \in \mathbb{Z}_q^{m\times m'} \mid ||\boldsymbol{B}_i|| \leq s\sqrt{m+m'}, 1 \leq i \leq r\}$.
  - Finally, $\mathcal{B}$ gives $(\boldsymbol{A}, \mathcal{M})$ to $\mathcal{A}$.
- **Breaking One-way Security**
  - $\mathcal{A}$ outputs $(X^*, \boldsymbol{B}^*, w^*_{\boldsymbol{B}} \in \mathbb{Z}^{m+m'})$ such that

$$X^* \subseteq \mathcal{M};\ \boldsymbol{B}^* \notin X^* \text{ and } \textbf{Verify}\left(\boldsymbol{B}^*, w^*_{\boldsymbol{B}}, \mathsf{Acc}_{X^*} \leftarrow \mathsf{Acc}(X^*)\right) = 1.$$

- **Solving SIS Instance**
  - $\textbf{Verify}(\boldsymbol{B}^*, w^*_{\boldsymbol{B}}, \mathsf{Acc}_{X^*}) = 1$ means

$$||\boldsymbol{B}^*|| \leq s\sqrt{m+m'},\ w^*_{\boldsymbol{B}} \in \Lambda^\perp(\boldsymbol{A}||\boldsymbol{A}\cdot(\mathsf{Acc}_{X^*} - \boldsymbol{B}^*)),\ \text{and } ||w^*_{\boldsymbol{B}}|| \leq s\sqrt{m+m'}$$

[4] To ensure that the $\mathsf{SIS}$ instance with norm bound $\beta = rs^3(m+m')$ is hard (worst-case to average-case reduction), the modulus $q$ of the scheme should satisfy $q > \beta \cdot w(\sqrt{n\log n})$ (See Theorem 1). In particular, for $q$ we choose the smallest prime bigger than $\lambda^t$ for the smallest $t$ such that $q > \beta\cdot\omega(\sqrt{n \log n})$. Choosing $n\log n$ for $\omega(\sqrt{n\log n})$, implies $\beta\cdot\omega(\sqrt{n\log n}) = \mathsf{poly}(\lambda)$, as $r, s, m, m', n$ are all bounded above by a $\mathsf{poly}(\lambda)$ size number.

– Let $\boldsymbol{C} = \mathsf{Acc}_{X^*} - \boldsymbol{B}^*$, and $w^*_{\boldsymbol{B}} = \begin{bmatrix} w^{*\prime}_{\boldsymbol{B}} \\ w^{*\prime\prime}_{\boldsymbol{B}} \end{bmatrix}$ such that $w^{*\prime}_{\boldsymbol{B}} \in \mathbb{Z}^m$, $w^{*\prime\prime}_{\boldsymbol{B}} \in \mathbb{Z}^{m'}$. Thus,

$$\boldsymbol{0} = (\boldsymbol{A}||\boldsymbol{AC})w^*_{\boldsymbol{B}} = \boldsymbol{A}(w^{*\prime}_{\boldsymbol{B}} + \boldsymbol{C}w^{*\prime\prime}_{\boldsymbol{B}})$$

– Finally, $\mathcal{B}$ outputs $\boldsymbol{e} = w^{*\prime}_{\boldsymbol{B}} + \boldsymbol{C}w^{*\prime\prime}_{\boldsymbol{B}}$ as a solution to $\mathsf{SIS}$ instance. The solution is valid as $\boldsymbol{e} \in \Lambda^{\perp}(\boldsymbol{A})$, and

$$\begin{aligned} ||\boldsymbol{e}|| &= ||w^{*\prime}_{\boldsymbol{B}} + \boldsymbol{C}w^{*\prime\prime}_{\boldsymbol{B}}|| \\ &\leq ||w^{*\prime}_{\boldsymbol{B}}|| + ||\boldsymbol{C}w^{*\prime\prime}_{\boldsymbol{B}}|| \\ &\leq ||w^{*\prime}_{\boldsymbol{B}}|| + ||w^{*\prime\prime}_{\boldsymbol{B}}||^2||\boldsymbol{C}|| \\ &\leq ||w^{*}_{\boldsymbol{B}}||^2||\mathsf{Acc}_{X^*} - \boldsymbol{B}^*|| \\ &\leq s^2(m+m')(rs\sqrt{m+m'}) \leq rs^3(m+m') = \beta. \end{aligned}$$

This Completes the proof.

# 4 Conclusion and Open Problems

We have provided the first lattice-based construction of a one-way accumulator scheme and proved its security from hardness assumption of the $\mathsf{SIS}$ problem (which is itself implied by worst-case lattice assumption). We leave open the problem of how to extend (our basic scheme) or construct a new lattice-based accumulator scheme with dynamic and universal functionalities. Another interesting problem is to extend our scheme such that zero-knowledge proofs of membership can be obtained.

# References


[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010. Full Version at "`http://crypto.stanford.edu/~dabo/pubs/papers/latticebb.pdf`".

[ABC+12] Jae Hyun Ahn, Dan Boneh, Jan Camenisch, Susan Hohenberger, Abhi Shelat, and Brent Waters. Computing on authenticated data. In *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2012.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In *ICALP*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999.

[AP09] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In *STACS*, volume 3 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[BdM93] Josh Cohen Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 274–285. Springer, 1993.

[BLL00] Ahto Buldas, Peeter Laud, and Helger Lipmaa. Accountable certificate management using undeniable attestations. In *ACM CCS*, pages 9–17, 2000.

[Boy10] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *PKC*, volume 6056 of *Lecture Notes in Computer Science*, pages 499–517. Springer, 2010.

[BP97] Niko Bari and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer, 1997.

[CF13] Dario Catalano and Dario Fiore. Vector commitments and their applications. In *PKC*, volume 7778 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2013.

[CHKO12] Philippe Camacho, Alejandro Hevia, Marcos Kiwi, and Roberto Opazo. Strong accumulators from collision-resistant hashing. *International Journal of Information Security*, 11(5):349–363, 2012.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.

[CJ10] Sébastien Canard and Amandine Jambert. On extended sanitizable signature schemes. In *CT-RSA*, volume 5985 of *Lecture Notes in Computer Science*, pages 179–194. Springer, 2010.

[CKS09] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In *PKC*, volume 5443 of *Lecture Notes in Computer Science*, pages 481–500. Springer, 2009.

[CL02] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2002.

[DKNS04] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.

[DT08] Ivan Damgård and Nikos Triandopoulos. Supporting non-membership proofs with bilinear-map accumulators. *IACR Cryptology ePrint Archive*, 2008:538, 2008.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. ACM, 2009.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *STOC 2008*, pages 197–206. ACM, 2008.

[LLX] Jiangtao Li, Ninghui Li, and Rui Xue. Universal accumulators with efficient non-membership proofs. In *ACNS*, Lecture Notes in Computer Science.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2006.

[LPRTj05] A. E. Litvak, A. Pajor, M. Rudelson, and N. Tomczak-jaegermann. Smallest singular value of random matrices and geometry of random polytopes. *Adv. Math*, 195:491–523, 2005.

[MGGR] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *IEEE Symposium on Security and Privacy*. IEEE Computer Society.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[Ngu05] Lan Nguyen. Accumulators from bilinear pairings and applications. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2005.

[Nyb96] Kaisa Nyberg. Fast accumulated hashing. In *FSE*, volume 1039 of *Lecture Notes in Computer Science*, pages 83–87. Springer, 1996.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 145–166. Springer, 2006.

[PS14] Henrich Christopher Pöhls and Kai Samelin. On updatable redactable signatures. In *ACNS*, volume 8479 of *Lecture Notes in Computer Science*, pages 457–475. Springer, 2014.

[PV08] Chris Peikert and Vinod Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 536–553. Springer, 2008.

[WWP08] Peishun Wang, Huaxiong Wang, and Josef Pieprzyk. Improvement of a dynamic accumulator at ICICS 07 and its application in multi-user keyword-based retrieval on encrypted data. In *APSCC*, pages 1381–1386. IEEE Computer Society, 2008.