Updatable Linear Map Commitments and Their Applications in Elementary Databases

Guiwen Luo
University of Waterloo
Waterloo, ON, Canada
guiwen.luo@uwaterloo.ca

Shihui Fu
University of Waterloo
Waterloo, ON, Canada
shihui.fu@uwaterloo.ca

Guang Gong, IEEE Fellow
University of Waterloo
Waterloo, ON, Canada
ggong@uwaterloo.ca

2021 18th International Conference on Privacy, Security and Trust (PST) | 978-1-6654-0184-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/PST52912.2021.9647740

***Abstract*—Linear map commitments allow the prover to commit to a vector, with the ability to prove the image of a linear map acting on the vector. In this paper, we propose linear map commitments with updatable feature and perfectly hiding property. Updatable feature means that the prover can update the commitment more efficiently than recompute the commitment when some of the entries in the committed vector are changed. Perfectly hiding property ensures the commitment reveals no information about the committed vector before opening. Then we present the implementation of our updatable linear map commitment (ULMC) over the 256-bit BN curve recommended in the SM9 standard, which provides around 100-bit security. The implementation shows that our ULMC schemes are efficient enough to support the elementary database constructions that simultaneously permit batching membership test, linear combination test, updatable feature and authenticity. Finally, we show that the ULMC-powered elementary databases are capable of supporting various applications where privacy and trust are the first priority such as exam result management systems, Internet of Things (IoT) management systems and business operations between banks and enterprises.**

***Index Terms*—Updatable linear map commitments, Internet of Things, Elementary databases.**

I. Introduction

Commitment schemes [1]–[5] are important cryptographic primitives which are widely used as a fundamental building block in a number of cryptographic protocols such as verifiable databases, cryptographic accumulators, elementary databases, secure multiparty computations, zero knowledge proofs, etc. A commitment scheme provides a tool to hide (i.e., Commit) a chosen value, with the property of being able to later reveal (i.e., Open) and check (i.e., Verify) the value or the result computed over the value. There are two properties to be considered, $binding$ and $hiding$. Informally speaking, binding means the committer cannot change the value after committing to it, and hiding means the commitment does not leak any information about the committed value.

Recently Lai and Malavolta introduced the notion of linear map commitments [6], where the committed value is a vector. Let ℓ, q be positive integers and $\mathbb{F}$ be a finite field. For a length-ℓ vector x and a linear map $f : \mathbb{F}^\ell \to \mathbb{F}^q$, a linear map commitment can be opened to $f(x)$. Linear map commitments are powerful enough to simultaneously support single/batching membership test and linear combination test, but their constructions of linear map commitments lack the updatable feature and hiding property.

One of the promising applications of linear map commitments is elementary databases [7]. Elementary databases are cryptographic tools that integrate with different commitments to provide different functionalities such as membership test, range test and so on [7], [8]. As increasing interest is focused on data security and privacy in modern cyber era, a number of applications call for versatile elementary databases.

An elementary database D is similar to a hash table, it is comprised of key-value pairs (x, y)'s such that each key x occurs only once and takes value $y = D(x)$. If we take the position index as the key, an elementary database allows one to commit a database that consists of the position-value pairs $\{(i, D(i))|1 \leq i \leq \ell, D(i) \in \mathbb{F}\}$ and prove that $D(i)$ is the value at position i.

In this paper, we propose the updatable feature and hiding property for linear map commitments, then employ the enhanced commitment schemes to construct elementary databases that achieve the following goals,

- *Single or batching membership test*. Given index set $I = \{i_1, ..., i_{|I|}\} \subseteq \{1, 2, ..., \ell\}$, test if alleged $\{D(i_1), ..., D(i_{|I|})\}$ are the values at positions I respectively.
- *Linear combination test*. Given index set $I = \{i_1, i_2, ..., i_{|I|}\} \subseteq \{1, 2, ..., \ell\}$ and vector $(a_{i_1}, a_{i_2}, ..., a_{i_{|I|}}) \in \mathbb{F}^{|I|}$, check if an alleged value is equal to $\sum_{i \in I} a_i D(i)$. Specially, check if an alleged value equals to the sum of $D(i)$'s or the average of $D(i)$'s.
- *Updatable feature*. When the original data is revised, the corresponding commitment can be efficiently updated.
- *Authenticity*. The elementary database should not be modified without the informed consent from users.

A. Related Work

Different commitments are proposed to serve different application scenarios [2]–[5], [9]. Here we mainly investigate the non-interactive vector commitments with trusted setup and their variants. The vector commitment, introduced by Catalano and Fiore [10], allows one to commit to a length-ℓ vector x, and later open at a specific index (i.e., prove that x_i is the i-th

TABLE I
COMPARISON OF THE PROPERTIES AMONG DIFFERENT COMMITMENTS.

	BMT	LCT	Updatable	Hiding
VC [10]	✓	✗	✓	✗
FC (Linear form) [11]	✗	✓	✗	✓
SVC [6]	✓	✗	✗	✗
LMC [6]	✓	✓	✗	✗
ULMC (This paper)	✓	✓	✓	✓

VC: Vector commitments, FC: Functional commitment, SVC: Subvector commitment, LMC: Linear map commitment, ULMC: Updatable linear map commitment. BMT: Batching membership test, LCT: Linear combination test.

entry of x). Then Libert *et al.* proposed a generalized notion of vector commitments called functional commitments [11]. A functional commitment allows one to commit to a value m and later reveal $f(m)$, i.e., the image of the function f computed over m. As an instantiation of functional commitments, linear form commitments are given in [11]. For a length-ℓ vector x and a linear form $f : \mathbb{F}^\ell \to \mathbb{F}$ $\big($for $f = (f_1, ..., f_\ell) \in \mathbb{F}^\ell, f(x) = \sum_{i=1}^{\ell} f_i x_i\big)$, a linear function commitment can later be opened to $f(x)$, thus enabling the linear combination test. Lai and Malavolta introduced subvector commitments and linear map commitments [6]. A subvector commitment is the batching version of the vector commitment, it allows one to committed to a length-ℓ vector x, and the commitment can be opened at an ordered index set $I \subseteq \{1, 2, ..., \ell\}$ $\big($i.e., prove $\{x_i, i \in I\}$ is the value in x at the indices I respectively$\big)$. linear map commitments are also instantiations of functional commitments, but they further generalized the subvector commitments and linear form commitments with respect to dimensions. For a length-ℓ vector x and a linear map $f : \mathbb{F}^\ell \to \mathbb{F}^q$, a linear map commitment can be opened to $f(x)$ $\big($for $f = (f_{ij}) \in \mathbb{F}^{q \times \ell}$, $f(x)$ is the result of the matrix f times the vector $x\big)$. Table I summarizes the properties of the aforementioned commitments.

For a length-ℓ vector x, if we compute a polynomial g by Lagrange interpolation over $\{(i, x_i)|i \in \{1, 2, ..., \ell\}\}$, then $g(i) = x_i$. Opening to the i-th entry of vector x is the same as opening to $g(i)$. In this sense, we can obtain vector commitments from polynomial commitments. Polynomial commitments, first proposed by Kate *et al.* [12], also developed into different variants that support committing to single polynomial or multiple polynomials, and opening to single point or multiple points on single polynomial or multiple polynomials [12]–[16].

The concept of elementary databases was introduced by Micali, Rabin and Kilian [7]. Their constructions support membership test and non-membership test. Chase *et al.* formalized a new kind of commitment scheme called mercurial commitments, then employed them as underlying building blocks to construct elementary databases [17]. Libert *et al.* [8] took the full advantage of mercurial commitments to construct elementary databases that support range queries over keys and values. Those constructions are equipped with zero-knowledge, but none of their constructions simultaneously permits batching membership test and linear combination test, which motivates our work.

B. Our Method and Contribution

In this paper, we mainly carry out our work in the following three aspects.

- *We propose the updatable feature and hiding property for linear map commitment schemes.* We need an algorithm to efficiently update the commitment if some entries of the committed vector are revised. By ditching the old exponents and multiplying the new exponents for every revised entry, we can compute the revised commitment more efficiently comparing to fully recompute it. The hiding property is achieved by adding an exponent to a random number. The method is similar to that in Pedersen commitment [18] and in functional commitments [11].
- *We implement the updatable linear map commitment over a 256-bit BN curve.* We analyze the complexity of the updatable linear map commitment schemes, then implement the schemes over system parameters of the 256-bit BN curve and R-ate pairing defined in SM9 [19], which provide around 100-bit security [20]. Our implementation is over asymmetric bilinear groups, thus providing higher efficiency compared with other symmetric setting counterparts.
- *We apply updatable linear map commitments to construct elementary databases that support batching membership test, linear combination test, updatable feature and authenticity.* To the best of our knowledge, this is the first elementary database scheme that simultaneously supports batching membership test, linear combination test and updatable feature. Suppose $f : \mathbb{F}^\ell \to \mathbb{F}^q$ is a linear map (f can be denoted as a matrix in $\mathbb{F}^{q \times \ell}$), $\{(i, D(i))|1 \leq i \leq \ell\}$ is a size-ℓ elementary database committed with an updatable linear map commitment. If one picks up f as a matrix whose each row is a standard unit vector (the vector with an entry 1 and the rest entries 0's), the elementary databases can achieve *single or batching membership test*. If one utilizes only the first row of f and set the rest rows as 0's, the elementary databases can achieve *linear combination test*. Finally, *updatable feature* and *authenticity* are inherently provided when employing updatable linear map commitment schemes. It is worth noting that the linear combination test can be used to query the sum or average of a subset of elementary database D. Suppose index set $I = \{i_1, i_2, ..., i_{|I|}\} \subseteq \{1, 2, ..., \ell\}$ and $D(I) := \{D(i)|i \in I\}$. One can query the sum of elements in $D(I)$ by taking the first row of the linear map f as vector $(a_1, a_2, ..., a_\ell)$ where $a_i = 1$ for $i \in I$ and the rest $a_i = 0$. One can further get the average of elements in $D(I)$ by dividing the sum by $|I|$.

C. Roadmap

The rest of the paper is organized as follows. In Section II, we introduce notations and some basic definitions. In Section III, we present the updatable linear map commitments with perfectly hiding property. In section IV, we analyze

the complexity of our commitment schemes, then show the implementation. In Section V, we apply our commitment schemes to construct the elementary databases. Finally, we conclude our paper and point out the future research direction in Section VI.

II. Preliminaries

Notations. Without special explanations, let D or key-value pairs $\{(x, D(x))\}$ be an elementary database, and $\{(i, D(i))|1 \leq i \leq \ell\}$ an elementary database whose key is the position index. Let $\ell \in \mathbb{N}$ be the size of a vector or an elementary database, $[\ell]$ the set $[\ell] := \{1, 2, ..., \ell\}$, p a big prime integer, and e a bilinear pairing defined over the elliptic curve groups. Let $\mathsf{poly}(\lambda)$ be a polynomial function in λ, $\epsilon(\lambda)$ a negligible function in λ, $y \leftarrow \mathbf{Alg}(x)$ an algorithm whose input is x and output is y, and $\alpha \leftarrow_{\$} \mathbb{Z}_p$ uniformly picking a random α from $\mathbb{Z}_p$.

A. Updatable Linear Map Commitments

Here we introduce updatable linear map commitments that empower our elementary databases. The following definition is derived from [6] by adding **UpdateCommit** algorithm.

Definition 1. (Updatable Linear Map Commitments (ULMC)). *An updatable linear map commitment is comprised of five algorithms* **ULMC = (Setup, Commit, UpdateCommit, Open, Verify)***:*

- $\mathsf{crs} \leftarrow$**Setup**$(1^\lambda, \mathcal{F}; \omega)$*: Let* $\ell, q \in \mathsf{poly}(\lambda)$ *be positive integers, and* $\mathcal{F} \subseteq \left\{f : \mathbb{F}^\ell \to \mathbb{F}^q\right\}$ *a family of linear maps, where* $\mathbb{F}$ *is the domain field. Taking the security parameter* 1^λ*, the description of the family* $\mathcal{F}$*, and a random tape* ω *as inputs, it generates and outputs a common reference string* crs*. We assume* crs *is input to following four algorithms, and do not write it out.*
- $(\mathsf{com}, \mathsf{aux}) \leftarrow$**Commit**$(x)$*: Taking a vector* $x \in \mathbb{F}^\ell$ *as input, it outputs a commitment string* com *and some auxiliary information* aux*.*
- $(\mathsf{com}', \mathsf{aux}') \leftarrow$**UpdateCommit**$(\mathsf{com}, \mathsf{aux}, x')$*: Taking the original commitment* com*, its corresponding* aux*, and the new vector* x' *as inputs, it outputs a revised commitment* com′ *to* x' *and a new auxiliary information* aux′*.*
- $\pi \leftarrow$**Open**(f, y, aux)*: Taking an* $f \in \mathcal{F}$ *, an image* $y \in \mathbb{F}^q$*, and some auxiliary information* aux *as inputs, it outputs a proof* π *showing that* $y = f(x)$*.*
- $b \in \{0, 1\} \leftarrow$**Verify**$(\mathsf{com}, f, y, \pi)$*: Taking a commitment string* com*, an* $f \in \mathcal{F}$*, an image* y*, and a proof* π *as inputs, it outputs 1, if and only if* com *is a commitment to* x *and* $y = f(x)$*.*

A ULMC scheme is said to be *correct* if an honest committer can convince the verifier, i.e., the probability that the verifier outputs 0 is negligible. A ULMC scheme is said to be *function binding*, if the PPT adversary cannot cheat using an inconsistent system. A ULMC scheme is said to be *perfectly hiding*, if the adversary cannot distinguish the commitments to two different vectors. Their formal definitions are given as follows,

Definition 2. (Function Binding). *An updatable linear map commitment over* $\mathbb{F}$ *is said to be function binding if for any PPT adversary* $\mathcal{A}$*, positive integers* $m, l, q \in \mathsf{poly}(\lambda)$*, and the set of linear maps* $\mathcal{F} \subseteq \left\{f : \mathbb{F}^\ell \to \mathbb{F}^q\right\}$*, the probability*

$$\Pr\left[\begin{array}{c|c} \forall k \in [m], f_k \in \mathcal{F} \wedge y_k \in \mathbb{F}^q \wedge & \mathsf{crs} \leftarrow \mathbf{Setup}\left(1^\lambda, \mathcal{F}; \omega\right) \\ \mathbf{Verify}\left(\mathsf{com}, f_k, y_k, \pi_k\right) = 1; & \left(\mathsf{com}, \{(f_k, y_k, \pi_k)\}_{k\in[m]}\right) \leftarrow \mathcal{A}(\mathsf{crs}) \\ \nexists x \in \mathbb{F}^\ell \text{ s.t. } \forall k \in [m], f_k(x) = y_k. & \end{array}\right]$$

is negligible in λ*.*

Definition 3. (Perfectly Hiding). *An updatable linear map commitment over* $\mathbb{F}$ *is said to be perfectly hiding if for a* crs *generated by an honest setup, for all vectors* $x_1, x_2 \in \mathbb{F}^\ell$ *with* $x_1 \neq x_2$*,* **Commit**(x_1) *and* **Commit**(x_2) *have identical distributions given that the random coins of* **Commit** *are uniformly randomly chosen.*

B. Elementary Databases

Elementary databases enable querying the database with a key and obtaining an answer to the problem whether there is a value associated to the key in the database. An elementary database D is similar to a hash table, it is comprised of key-value pairs (x, y)'s such that each key x occurs only once and takes value $y = D(x)$. This paper takes the position indices as the keys of an elementary database D, thus it allows one to commit a database that consists of the position-value pairs $(i, D(i))'s$ and prove that $D(i)$ is the value at position i.

Definition 4. (Elementary Databases (EDB)). *An elementary database consists of the following four algorithms* **EDB=(Setup, CommitDB, Prove, Verify)***:*

- $\mathsf{crs} \leftarrow$**Setup**$\left(1^\lambda, \ell; \omega\right)$*: Taking security parameter* 1^λ*, database size* ℓ *and random tape* ω *as inputs, it generates and outputs common reference string* crs*. We assume* crs *is implicitly input to following three algorithms.*
- $(\mathsf{com}, \mathsf{aux}) \leftarrow$**CommitDB**$(D)$*: Taking the database* D *as input, it outputs the commitment to* D*,* com*, and auxiliary information* aux*.*
- $\pi_x \leftarrow$**Prove**(x, y, aux)*: Taking the key* x*, value* y*, and auxiliary information* aux *as inputs, it outputs a proof* π_x *that* $y = D(x)$ *in this database.*
- $\{0, 1\} \leftarrow$**Verify** $(\mathsf{com}, x, y, \pi_x)$*: Taking the database commitment* com*, key* x*, value* y*, proof* π_x *as inputs, it outputs 1 if and only if* com *is a commitment to* D *and* $y = D(x)$ *in the database.*

Two security properties are of interest, *completeness* and *soundness*. Completeness implies that an honest prover with an honestly generated proof can always convince a verifier. Soundness guarantees that the prover cannot cheat a verifier with a forged keyed value $D(x)$.

Some other EDB constructions such as [7], [8], [17] may consider *zero-knowledge* property that ensures the protocol to prove $D(x)$ is the value with respect to x without revealing any further information about the database D, even without

leaking the size of D [7], [8]. Our EDB construction in this paper is not zero-knowledge.

III. Updatable linear map commitments with perfectly hiding property

Our ULMC construction is inspired by the scheme presented in [6] via adding the updatable feature and perfectly hiding property. Those two properties are necessary in various scenarios. For instances, in an examination where false marks are given, the exam organizers should have the ability to revise the marks with the consent of the exam participants. Or in an IoT network where a sensor generates wrong data, there should be a way to correct it and efficiently update the commitment. For all scenarios we do not want the commitment to leak any information about the database before opening process, thus the perfectly hiding property fits in.

The construction is presented over asymmetric pairing groups in order to achieve better efficiency, which is the case of our implementation in Section IV.

Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an asymmetric elliptic curve bilinear pairing, prime integer p the order of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, and G, H the generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively. Let x be a vector in $\mathbb{Z}_p^\ell$, $q \in \mathbb{N}$ the dimension of the image and $\mathcal{F} := \{f | f : \mathbb{Z}_p^\ell \to \mathbb{Z}_p^q\}$ the set consisting of all linear maps from $\mathbb{Z}_p^\ell$ to $\mathbb{Z}_p^q$. A ULMC instance defined over asymmetric bilinear pairing is comprised of five algorithms **ULMC = (Setup, Commit, UpdateCommit, Open, Verify)**:

- $\mathsf{crs} \leftarrow$**Setup**$(1^\lambda, \mathcal{F}; \omega)$:
  - $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, G, H, e) \leftarrow \mathrm{GGen}\left(1^\lambda; \omega\right)$
  - $\alpha, z_1, \ldots, z_q \leftarrow_\$ \mathbb{Z}_p$
  - $\forall j \in [\ell], G_j := G^{\alpha^j}$,
    $\forall i \in [q], j \in [2\ell] \backslash \{\ell+1\}$,
    $G_{i,j} := G^{z_i \alpha^j}, H_{i,j} := H^{z_i \alpha^j}$,
    $\mathsf{crs} := \left(\begin{array}{c} p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, G, H, \{G_j\}_{j \in [\ell]}, \\ \{G_{i,j}, H_{i,j}\}_{i \in [q], j \in [2\ell] \backslash \{\ell+1\}}, e \end{array} \right)$.
    return crs
- $(\mathsf{com}, \mathsf{aux}) \leftarrow$**Commit**$(x)$:
  - Suppose vector $x = (x_1, x_2, ..., x_\ell)$ is of length ℓ. Compute
    $\mathsf{com} := G^\gamma \cdot \prod_{j \in [\ell]} G_j^{x_j}$ for $\gamma \leftarrow_\$ \mathbb{Z}_p$,
    $\mathsf{aux} := (x, \gamma)$.
    return $(\mathsf{com}, \mathsf{aux})$
- $(\mathsf{com}', \mathsf{aux}') \leftarrow$**UpdateCommit**$(\mathsf{com}, \mathsf{aux}; x')$:
  - Parse aux as (x, γ). Suppose x' is also of length ℓ. Let $I_{x-x'} := \{i \,| x_i \in x \text{ and } x_i \notin x'\}$ be an index set consisting of all entries in x but not in x'. Compute
    $$\mathsf{com}' := \mathsf{com} \cdot G^{\gamma'} \cdot \prod_{i \in I_{x-x'}} G_i^{-x_i} \cdot \prod_{i \in I_{x-x'}} G_i^{x_i'}$$
    for a randomly chosen $\gamma' \leftarrow_\$ \mathbb{Z}_p$,
    $\mathsf{aux}' := (x', \gamma + \gamma')$.
    return $(\mathsf{com}', \mathsf{aux}')$
- $\pi \leftarrow$**Open**(f, y, aux) :
  - Parse aux as (x, γ),
    $$\pi := \prod_{i \in [q]} \prod_{j \in [\ell]} \left(G_{i,\ell+1-j}^{\gamma} \cdot \prod_{k \in [\ell] \backslash \{j\}} G_{i,\ell+1+k-j}^{x_k} \right)^{f_{i,j}}.$$
    return π
- $b \in \{0,1\} \leftarrow$**Verify**$(\mathsf{com}, f, y, \pi)$:
  - Check if $y \in \mathbb{Z}_p^\ell$;
  - Check if the following equation holds,
    $$e\left(\mathsf{com}, \prod_{i \in [q]} \prod_{j \in [\ell]} H_{i,\ell+1-j}^{f_{i,j}}\right) = e\left(G_1, \prod_{i \in [q]} H_{i,\ell}^{y_i}\right) \cdot e(\pi, H).$$

    If both checks pass, output 1. Otherwise output 0.

The updatable property is inspired by [10], [21]. One can see that **UpdateCommit** is more efficient than directly call **Commit** to re-compute the commitment string when the size of the revised entries is less than half of that of the entire committed vector. In **UpdateCommit**, there is no need to multiply the term $G^{-\gamma}$, because the resulting secret hint will be $\gamma + \gamma'$, which is also random.

Theorem 1. *Suppose $\ell, q \in \mathsf{poly}(\lambda)$, and suppose $1/p \in \epsilon(\lambda)$ is negligible in λ. The above ULMC is function binding in the generic bilinear group model.*

Proof. The proof is given in the full version due to the page limitation. Essentially the proof is similar to the one presented in [6]. □

Theorem 2. *The above ULMC is perfectly hiding.*

Proof. Notice that for any fix vector x, when γ is uniformly randomly chosen from $\mathbb{Z}_p$, the resulting commitment com is uniformly distributed over $\mathbb{G}_1$. This implies that for any vector x, the random variables describing the output of **Commit** are identically distributed, which indicates that the ULMC is perfectly hiding. □

IV. Complexity Analysis and Implementation

In this section, we analysis the complexity of our ULMC presented in Section III, then provide an implementation in C++ to illustrate its efficiency.

Suppose $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is the asymmetric pairing used in our ULMC. Three essential operations involved in ULMC are addition, scalar multiplication and asymmetric pairing computation over elliptic curve groups. Let A_1, A_2 be the addition operation in G_1, G_2 respectively, M_1, M_2 the scalar multiplication in G_1, G_2 respectively, and E the R-ate pairing computation. Let x be a vector of length ℓ, $I = \{i_1, i_2, ..., i_{|I|}\} \subseteq [\ell]$ and $x_I := \{x_i | i \in I\}$.

TABLE II
COMPLEXITY OF DIFFERENT OPERATIONS

	Commit	UpdateCommit	Open	Verify
BMT	$(\ell+1)M_1$	$(2\|I\|+1)M_1$	$\|I\|(\ell+1)M_1$	$3E+\|I\|M_2$ $+(2\|I\|-2)A_2$
LCT	$(\ell+1)M_1$	$(2\|I\|+1)M_1$	$\|I\|(\ell+1)M_1$	$3E+(\|I\|+1)M_2$ $+(\|I\|-1)A_2$
ST	$(\ell+1)M_1$	$(2\|I\|+1)M_1$	$\|I\|\ell M_1$	$3E+M_2$ $+(\|I\|-1)A_2$

Complexity of batching membership test (BMT), linear combination test (LCT) and sum test (ST) corresponding to index set $I \subset [\ell]$. We take $q = \ell$ so that all the data in the committed vector can support membership test. Sum test is for the case when the coefficients of linear combination are 1's, i.e., test if an alleged value equals to $\sum_{i\in I} D(i)$.

TABLE III
TIMING OF DIFFERENT OPERATIONS

		Commit	Update	Open	Verify
$\|I\| = 100$, $\ell = 100$	BMT	0.10	NA	10.18	0.17
	LCT				0.17
	ST			10.09	0.01
$\|I\| = 100$, $\ell = 1,000$	BMT	1.01	0.20	100.60	0.17
	LCT				0.17
	ST			100.51	0.01
$\|I\| = 100$, $\ell = 10,000$	BMT	10.05	0.20	1008.23	0.17
	LCT				0.17
	ST			1006.97	0.01

Unit: Second. Timing of batching membership test (BMT), linear combination test (LCT) and sum test (ST) for $|I| = 100$ and different ℓ's. Time is measured on a 2.6 GHz processor running single thread.

A. Complexity Analysis

According to the ULMC scheme presented in Section III, we obtain the complexity of different algorithms as follows.

- **Commit.** The complexity is $(\ell+1)M_1 + \ell A_1$.
- **UpdateCommit.** Let I be the index set indicating revised entries, then its complexity is $(2|I|+1)(M_1+A_1)$.
- **Open.** In batching membership test, f is a matrix where each row is a standard unit vector (the vector with an entry 1 and the rest entries 0's). Let I be the index set of entries to be tested, then $f_{i,j} = 1$ when $i = j$ for $i, j \in I$, $f_{i,j} = 0$ for the rest indices. The complexity is $|I|(\ell+1)M_1 + (|I|\ell - 1)A_1$. In linear combination test, only the first row of f is non-zero, i.e., $f_{i,j} \neq 0$ for $i = 1, j \in I$, $f_{i,j} = 0$ for the rest indices. The complexity is $|I|(\ell+1)M_1 + (|I|\ell - 1)A_1$. For sum test, the functionality is testing if an alleged value equals to $\sum_{i\in I} x_i$, which is a special case of linear combination test with $f_{i,j} = 1$ for $i = 1, j \in I$. The complexity is $|I|\ell M_1 + (|I|\ell - 1)A_1$.
- **Verify.** Note that if $f_{i,j} = 1$, we can get $H_{i,\ell+1-j}^{f_{i,j}}$ for free. Following similar analysis as that of Prove, we can obtain that the Verify complexity of batching membership test is $3E + |I|M_2 + (2|I|-2)A_2$. In linear combination test, $y_i = 0$ for $2 \leq i \leq \ell$, so the Verify complexity is $3E + (|I|+1)M_2 + (|I|-1)A_2$. The Verify complexity of sum test is $3E + M_2 + (|I|-1)A_2$.

In practice, the order of G_1, G_2 is usually no less than 256-bit, which indicates that $A_i/M_i \leq 0.005, i = 1, 2$. We omit A_i in **Commit**, **UpdateCommit** and **Open**. The result is summarized in Table II. We have $\ell \geq |I| \geq 2$ in batching membership test, linear combination test and sum test, under this circumstance Table II shows that the performance bottleneck is **Open**.

B. Implementation

We implement our ULMC scheme over the 256-bit BN curve and R-ate pairing recommended in the SM9 standard [19]. The BN curve is defined by equation $E : y^2 = x^3 + 5$ over the finite field $\mathbb{F}$ with characteristic

```
char  =  0xB6400000 02A3A6F1 D603AB4F F58EC745
         21F2934B 1A7AEEDB E56F9B27 E351457D,
```

and the group order of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ is

```
order  =  0xB6400000 02A3A6F1 D603AB4F F58EC744
          49F2934B 18EA8BEE E56EE19C D69ECF25.
```

In our implementation, we assume the basic operations (addition, subtraction, multiplication and inversion) in $\mathbb{F}$ have been implemented, we employ the multiple precision arithmetic library GMP to achieve this[1]. We implement degree 12 extension of $\mathbb{F}$ as a "tower" of subfield extensions in order to efficiently compute R-ate pairing e. We implement additions, scalar multiplications over elliptic curve groups $\mathbb{G}_1, \mathbb{G}_2$ and R-ate pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Finally we implement ULMC scheme and test its performance for different $|I|$'s and ℓ's. The result is presented in Table III. When $|I| \geq \ell/2$, it is more efficient to directly invoke **Commit** rather than **UpdateCommit**.

V. ULMC-POWERED ELEMENTARY DATABASES AND THEIR APPLICATIONS

We build our elementary database $D = \{(i, D(i)),\ 1 \leq i \leq \ell\}$ by integrating D with ULMC scheme. Such an EDB extends the basic EDB presented Definition 4 in two aspects:

- It has an updatable feature.
- It provides single/batching membership test and linear combination test.

Security properties of the EDB construction are ensured. *Completeness* of the elementary database is trivial. By following the scheme, one can easily conclude that the verification will pass. *Soundness* is derived from the function binding property of the commitment in the generic bilinear group model. If $y \neq f(D)$ and the verification passes with probability that is non-negligible, it would be a direct contradiction to the function binding property.

We finally propose three scenarios where our EDB's can be deployed.

- In the situation where participants are taking a college entrance exam that may determine their future. The exam is so important that all the participants want to have such an elementary database powered exam result management

[1]GMP: The GNU Multiple Precision Arithmetic Library, https://gmplib.org/

system that could ensure them to get trustworthy results and the exam organizers could not manipulate the results under the table.
- In the case that a nuclear power plant is cracked. Authorities set up a nuclear leakage monitoring system that consists of network of sensors. The authorities allege the environment to be secure but the residents do not trust it, so it is eagerly needed to have such an elementary database powered Internet of Things (IoT) management system that enables to query the average of all sensor data in a trustworthy way and that make sure the authorities do not have the ability to modify the data behind the scene.
- In the circumstance where a bank for some business reasons requires the sum of employees' wages of a company and the company wants to keep individual employee's data secret while convinces the bank, it needs an elementary database powered wage management system that can check if alleged value equals to the sum of the wages without revealing individual wages.

All above application scenarios do not require zero-knowledge, because after the opening process, an exam participant is supposed to know his marks, a citizen is supposed to know the average of the monitoring data in the nuclear leakage monitoring system, and a bank is supposed to know the sum of the wages. The implementation benchmark in Table III shows that our newly constructed EDB's are capable of supporting our envisioned application scenarios.

- In exam result management systems, an exam participant wants to query their individual results, and a class coordinator wants to query the average mark of the class. In this situation $|I|$ is the class size, and it is reasonable to take $|I| = 100$. Table III shows that our EDB can support an exam result management system whose size is 10,000 with reasonable Prove time in about 1,000 seconds.
- In IoT management systems or support of the business between banks and enterprises, the sum of all entries in the EDB or the average of all entries in the EDB are usually needed, which means $|I| = \ell$, where ℓ is the size of the IoT system or the enterprise. Table III shows that our EDB can support such a system whose size is 100 with Prove time in around 10 seconds.

VI. Conclusion

With the increasing interest to data security and privacy, applications such as exam result management systems, Internet of Things and empowering business between banks and enterprises call for versatile elementary databases.

In this paper we proposed the updatable feature and hiding property for linear map commitment schemes to obtain ULMC scheme, then we analyzed the complexity of our ULMC scheme and presented the implementation. We applied our ULMC to construct the first ever elementary databases that simultaneously support batching membership test, linear combination test, updatable feature and authenticity, thus fulfilling the goal to support aforementioned applications.

Next, we will strive for further speeding up our implementation so that our EDB's can be deployed to applications with larger scales.

Acknowledgement

The work is supported by NSERC SPG grant.

References

[1] M. Blum, "Coin flipping by telephone a protocol for solving impossible problems," *ACM SIGACT News*, vol. 15, no. 1, pp. 23–27, 1983.
[2] G. Brassard, D. Chaum, and C. Crépeau, "Minimum disclosure proofs of knowledge," *Journal of computer and system sciences*, vol. 37, no. 2, pp. 156–189, 1988.
[3] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems," *Journal of the ACM (JACM)*, vol. 38, no. 3, pp. 690–728, 1991.
[4] M. Yung and R. Impagliazzo, "Direct minimum-knowledge computations," in *Proc. Crypto*, vol. 87, 1998, pp. 40–51.
[5] I. Damgård, "Commitment schemes and zero-knowledge protocols," in *School organized by the European Educational Forum*. Springer, 1998, pp. 63–86.
[6] R. W. Lai and G. Malavolta, "Subvector commitments with application to succinct arguments," in *Annual International Cryptology Conference*. Springer, 2019, pp. 530–560.
[7] S. Micali, M. Rabin, and J. Kilian, "Zero-knowledge sets," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE, 2003, pp. 80–91.
[8] B. Libert, K. Nguyen, B. H. M. Tan, and H. Wang, "Zero-knowledge elementary databases with more expressive queries," in *IACR International Workshop on Public Key Cryptography*. Springer, 2019, pp. 255–285.
[9] M. Naor, "Bit commitment using pseudorandomness," *Journal of cryptology*, vol. 4, no. 2, pp. 151–158, 1991.
[10] D. Catalano and D. Fiore, "Vector commitments and their applications," in *International Workshop on Public Key Cryptography*. Springer, 2013, pp. 55–72.
[11] B. Libert, S. Ramanna, and M. Yung, "Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions," in *43rd International Colloquium on Automata, Languages and Programming (ICALP 2016)*, 2016.
[12] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *International conference on the theory and application of cryptology and information security*. Springer, 2010, pp. 177–194.
[13] A. Gabizon, Z. J. Williamson, and O. Ciobotaru, "Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 953, 2019.
[14] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn, "Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2111–2128.
[15] D. Boneh, J. Drake, B. Fisch, A. Gabizon, and A. Protocol, "Efficient polynomial commitment schemes for multiple points and polynomials." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 81, 2020.
[16] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, and N. Ward, "Marlin: Preprocessing zksnarks with universal and updatable srs," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Cham, 2020, pp. 738–768.
[17] M. Chase, A. Healy, A. Lysyanskaya, T. Malkin, and L. Reyzin, "Mercurial commitments with applications to zero-knowledge sets," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2005, pp. 422–439.
[18] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual international cryptology conference*. Springer, 1991, pp. 129–140.
[19] "GM/T 0044.5 - 2016: Identity-based cryptographic algorithms SM9 - Part 5: Parameter definition," Standard, 2016.
[20] R. Barbulescu, N. El Mrabet, and L. Ghammam, "A taxonomy of pairings, their security, their complexity," 2019.
[21] S. Fu, G. Luo, and G. Gong, "Batching anonymous and non-anonymous membership proofs for blockchain applications," in *2021 IEEE World AI IoT Congress (AIIoT)*. IEEE, 2021, pp. 0342–0348.