Functional Commitment Schemes: From Polynomial Commitments to Pairing-Based Accumulators from Simple Assumptions[*]

Benoît Libert[†][1], Somindu C. Ramanna[‡][2], and Moti Yung[§][3]

1 ENS de Lyon, LIP Laboratory, Lyon, France
benoit.libert@ens-lyon.fr

2 ENS de Lyon, LIP Laboratory, Lyon, France
somindu.ramanna@ens-lyon.fr

3 Snapchat, Los Angeles, CA, USA; and
Columbia University, New York, USA
moti@cs.columbia.edu

Abstract

We formalize a cryptographic primitive called functional commitment (FC) which can be viewed as a generalization of vector commitments (VCs), polynomial commitments and many other special kinds of commitment schemes. A non-interactive functional commitment allows committing to a message in such a way that the committer has the flexibility of only revealing a function of the committed message during the opening phase. We provide constructions for the functionality of linear functions, where messages consist of vectors over some domain and commitments can later be opened to a specific linear function of the vector coordinates. An opening for a function thus generates a witness for the fact that the function indeed evaluates to a given value for the committed message. One security requirement is called *function binding* and requires that no adversary be able to open a commitment to two different evaluations for the same function.

We propose a construction of functional commitment for linear functions based on constant-size assumptions in composite order groups endowed with a bilinear map. The construction has commitments and openings of constant size (i.e., independent of n or function description) and is *perfectly hiding* – the underlying message is information theoretically hidden. Our security proofs build on the Déjà Q framework of Chase and Meiklejohn (Eurocrypt 2014) and its extension by Wee (TCC 2016) to encryption primitives, thus relying on constant-size subgroup decisional assumptions. We show that FC for linear functions are sufficiently powerful to solve four open problems. They, first, imply polynomial commitments, and, then, give cryptographic accumulators (i.e., an algebraic hash function which makes it possible to efficiently prove that some input belongs to a hashed set). In particular, specializing our FC construction leads to the first pairing-based polynomial commitments and accumulators for large universes known to achieve security under simple assumptions. We also substantially extend our pairing-based accumulator to handle subset queries which requires a non-trivial extension of the Déjà Q framework.

1998 ACM Subject Classification E.3 Data Encryption, K.6.5 Security and Protection

[*] Full version available at https://hal.inria.fr/hal-01306152.
[†] The first author was funded by the "Programme Avenir Lyon Saint-Etienne de l'Université de Lyon" in the framework of the programme "Investissements d'Avenir" (ANR-11-IDEX-0007).
[‡] The second author was funded by the "Programme Avenir Lyon Saint-Etienne de l'Université de Lyon" in the framework of the programme "Investissements d'Avenir" (ANR-11-IDEX-0007).
[§] Part of this work was done while the third author was with Google Inc. and visiting the Simons Institute for Theory of Computing at U.C. Berkeley.

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 30; pp. 30:1–30:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases Cryptography, commitment schemes, functional commitments, accumulators, provable security, pairing-based, simple assumptions

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.30

1 Introduction

Commitment schemes are fundamental primitives used as building blocks in a number of cryptographic protocols. A commitment scheme emulates a publicly observed safe; it allows a party to commit to a message m so that this message is not revealed until a later moment when the commitment is opened and the receiver gets convinced that the message was indeed m. Two important security properties are called hiding and binding. The former requires that no information about the message is revealed to an observer. The latter property means that the committing party cannot change the message after committing to it.

Several works considered commitment schemes where the committer has the flexibility of only revealing some partial information about the message (rather than the entire message) during the opening phase. In vector commitments [22, 10], messages are *vectors* and commitments are only opened with respect to specific positions. Another example is polynomial commitments, where users commit to a polynomial and only reveal evaluations of this polynomial on certain inputs.

In this work, we consider functional commitments (FC) for linear functions. Namely, messages consist of vectors $(m_1, \ldots, m_n)$ and commitments can be partially opened by having the sender verifiably reveal a linear combination $\sum_{i=1}^{n} x_i \cdot m_i$, for public coefficients $\{x_i\}_{i=1}^{n}$. We show that this functionality implies many other natural functionalities, including vector commitments, polynomial commitments and cryptographic accumulators. We provide an efficient FC realization for linear functions based on well-studied assumptions in groups with a bilinear map. In turn, our scheme implies solutions to past natural questions. We give the first constructions under constant-size assumptions of two important primitives: polynomial commitments and cryptographic accumulators. In both cases, earlier solutions were based on non-standard assumptions where the number of input elements (and thus the strength of the assumption) depended on specific features of the schemes (like the maximal degree of committed polynomials). Our third result is a solution to an accumulator supporting subset queries, which is also based on constant size assumptions.

1.1 Related Works and the Open Problems

Functional commitments. Functional commitments can be seen as the natural commitment analogue of functional encryption [31, 6]. The latter primitive allows restricting what the receiver learns about encrypted data: when decrypting using a secret key SK_F for the function F, the decryptor learns $F(x)$ and nothing else. Likewise, FC schemes allow the committer to accurately control what the opening phase can reveal about the message.

Functional commitments were implicitly suggested by Gorbunov, Vaikuntanathan and Wichs [17] who described a statistically-hiding commitment scheme for which the sender is able to only reveal a circuit evaluation $C(x)$ when x is the committed input. While their solution supports arbitrary circuits and relies on well-studied lattice assumptions, its input x must be committed to in a bit-by-bit manner (or at least by splitting x into small blocks). We remark that, assuming a common reference string, non-interactive FC for general functionalities can be realized by combining ordinary statistically-hiding commitments with non-interactive zero-knowledge (NIZK) proofs [3]. Here, we focus on the problem of achieving a better efficiency for more restricted (yet, sufficiently powerful for many applications) functionalities. Assuming a common reference string (as in all non-interactive perfectly hiding commitments), we aim at efficient constructions supporting short witnesses without resorting to the machinery of NIZK proofs. In particular, we aim at constant-size commitment strings (regardless of how long the message is) supporting concise witnesses.

In the literature, a number of earlier works consider settings where a sender is given the flexibility of revealing only a partial information about committed data. A verifiable random function [25], for example, can be seen as a perfectly binding commitment to a pseudo-random function key for which the committer can convince a verifier about the correct function evaluation for the committed key on a given input. Selective-opening security [16] addresses the problem of proving the security of un-opened commitments when an adversary gets to see the opening of other commitments to possibly correlated messages.

Zero-knowledge sets, as introduced by Micali, Rabin and Kilian [24], are another prominent example where users commit to a set S or an elementary database and subsequently prove the (non-)membership of some elements without revealing any further information (not even the cardinality of the committed set S). Ostrovsky, Rackoff and Smith [27] envisioned committed databases for which the sender can demonstrate more general statements than just membership and non-membership.

Vector commitments. Concise vector commitments were first suggested by Libert and Yung [22] and further developed by Catalano and Fiore [10]. They basically consist of Pedersen-like [30] commitments to vectors $(m_1, \ldots, m_n)$ where a constant-size opening (where "constant" means independent of n) allows the sender to open the commitment for only one coordinate m_i without revealing anything on other coordinates. The initial motivation of vector commitments was the design of zero-knowledge databases with short proofs [11, 22] via mercurial commitments [12] supporting short coordinate-wise openings [22]. While concise vector commitments can be based on long-lived hardness assumptions like RSA or Computational Diffie-Hellman [10], they either require groups of hidden order or public keys of size $O(n^2)$ if n is the dimension of committed vectors. In contrast, solutions based on variable-size assumptions allow for public keys of size $O(n)$, which leaves open the following problem.

Problem 1: *Is there a concise vector commitment scheme achieving linear-size public keys under constant-size assumptions in groups with a bilinear map?*

Polynomial commitments. As introduced by Kate, Zaverucha and Goldberg [19], polynomial commitments are a mechanism whereby a sender can generate a constant-size commitment to a polynomial $P[Z]$ (where "constant" means independent of the degree) in such a way that a constant-size witness can convince a verifier that the committed $P[Z]$ indeed evaluates to $P(i)$ for a given i. Polynomial commitments find natural applications in the context of verifiable secret sharing [14], anonymous credentials with attributes [7] or in optimized flavors of zero-knowledge databases which do not seek to hide the size of the committed set. They also imply vector commitments, as observed in [7]. Camenisch *et al.* [7] used vector commitments in a modular design of anonymous credentials where users' credentials are associated with descriptive attributes. While the commitments in [19, 7] were based on parameterized assumptions, the problem described below has been open.

Problem 2: *Design a polynomial commitment based on constant-size assumptions.*

Accumulators. Cryptographic accumulators can be interpreted as commitments, especially when the hashing algorithm is randomized. Accumulators [2] are closely related to zero-knowledge sets in that they make it possible to hash a set S while efficiently generating witnesses guaranteeing the inclusion of certain elements in the hashed set. Unlike zero-knowledge sets, they do not hide the cardinality of the underlying set but usually achieve a better efficiency via short membership witnesses. The first family of accumulators based on number theoretic techniques relies on groups of hidden order [2, 1, 23, 4] and includes proposals based on the Strong RSA assumption [1, 21]. The second family [26, 8], which was first explored by Nguyen [26], appeals to bilinear maps (a.k.a. pairings) and assumptions whose hardness depends on a parameter q determined by features of the scheme or the number of adversarial queries.

Solutions based on the Strong RSA assumption feature short public parameters and readily extend into universal accumulators [21] (where non-membership witnesses can show that a given input was not accumulated) or dynamic accumulators [9] (where witnesses can be autonomously updated when the hashed set is modified). On the other hand, they usually require expensive operations to injectively encode set elements as prime numbers. While pairing-based schemes [26, 8] do not need such a prime-number-encoding, they require linear-size public parameters in the maximal number of accumulated elements. On the positive side, they are useful in applications where the number of hashed elements cannot exceed a pre-determined bound. Pairing-based accumulators also proved useful in the context of authenticated data structures. Papamanthou *et al.* [29] used them to authenticate set operations and notably prove (using a constant-size witness) the inclusion of a given set in the accumulated set. The same technique was extended [29] to provide evidence that two accumulated sets have a given intersection.

A third family of accumulators [28, 4] builds on hash trees rather than number theoretic assumptions. Its disadvantage is that witnesses have size $O(\log N)$ (where N denote the cardinality of hashed sets) whereas number-theoretic solutions enable $O(1)$-size witnesses.

The security properties of accumulators were recently re-formalized by Derler *et al.* [15] who showed connections with other primitives. It was notably showed that, when endowed with an indistinguishability property, accumulators imply non-interactive commitment schemes and are implied by zero-knowledge sets.

Despite their numerous applications, cryptographic accumulators still have relatively few assumptions to rely on. So far, known candidates based on standard assumption arise from a generic construction from vector commitments [10]. While implying solutions based on RSA or Diffie-Hellman, the generic construction of [10] only supports inputs living in a small domain: the public key size is indeed linear in the size of the input universe, which prevents from hashing elements consisting of arbitrary strings. This leaves open Problem 3.

Problem 3: *Does there exist a pairing-based accumulator for large input universes secure under constant-size assumptions?*

As mentioned earlier, accumulators are applicable in authenticating set operations ([29]) and a useful extension would allow creating witnesses for set inclusion and intersection that are of constant size. Namely, a short witness can serve as evidence that some set X is a subset of the accumulated set or that two sets X_1, X_2 have a particular intersection I. In this domain, the following problem still remains open.

Problem 4: *Construct a pairing-based accumulator supporting set operations with constant-size witnesses achieving security under simple assumptions.*

1.2 Our Contributions

We formalize the notion of *functional commitments* (FCs) for linear functions, a generalization of vector commitments (VCs). Similar to VCs, such a commitment scheme allows committing to vectors of messages which can later be opened to specific function evaluations. While possible [17], the design of FCs for arbitrary functionalities seems unlikely to lead to truly efficient solutions. Instead, we aim at FCs for linear function families $\{F_{\vec{x}} : \mathcal{D}^n \times \mathcal{D}^n \to \mathcal{D}\}_{\vec{x} \in \mathcal{D}^n}$ defined by $F_{\vec{x}}(\vec{m}) = \langle \vec{x}, \vec{m} \rangle = \sum_{i=1}^{n} x_i m_i$ for $\vec{m} \in \mathcal{D}^n$ that suffice for many important applications. An FC scheme for a family of linear functions $\{F_{\vec{x}} : \mathcal{D}^n \to \mathcal{D}\}_{\vec{x} \in \mathcal{D}^n}$ produces commitments to messages of the form $\vec{m} = (m_1, \ldots, m_n) \in \mathcal{D}^n$ over the domain $\mathcal{D}$. Fixing a specific $\vec{x} \in \mathcal{D}^n$, such that $y = \sum_{i=1}^{n} x_i m_i \in \mathcal{D}$, an opening for $F_{\vec{x}}$ demonstrates that $F_{\vec{x}}(\vec{m})$ indeed evaluates to y. The security notions of hiding and binding extend to our setting in a natural way. In addition, we require the commitments and witnesses to be *concise* i.e., their size should be independent of the length of messages or function description.

Our first contribution is a construction of functional commitment for linear functions based on well-studied assumptions in composite order bilinear groups. The scheme is perfectly hiding and computationally binding under subgroup decision assumptions. The construction can be seen as a variant of the vector commitment scheme of Izabachène *et al.* [18] which was only proved secure under a non-standard variable-size assumption. We show that the composite-order setting makes it possible to use the Déjà Q framework of [13] so as to obtain security from constant size assumptions. As FC for linear functions implies vector commitments, our construction provides a positive answer to *Problem 1*.

As a second contribution, we show that our FC scheme implies polynomial commitments and large-universe accumulators supporting subset queries. The resulting schemes are secure under subgroup decision assumptions of constant-size thus settling *Problem 2* and *Problem 3*. We finally extend our accumulator into a scheme supporting subset queries while retaining security from constant size assumptions, partially answering *Problem 4* in the affirmative.

Overview of our Construction. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map with common group order $N = p_1 p_2 p_3$ and let $\mathbb{G}_q$ denote the subgroup of $\mathbb{G}$ of order q (here q would be of the form $p_1^{e_1} p_2^{e_2} p_3^{e_3}$ for $e_1, e_2, e_3 \in \{0, 1\}$). The linear functions will be defined over $\mathbb{Z}_N$. The commitment key consists of $\{g^{\alpha^j}\}_{j=1}^{n}$, $\{U_j = u^{\alpha^j}\}_{j \in [1,2n] \setminus \{n+1\}}$ for some $g, u \in \mathbb{G}_{p_1}$. The trapdoor is $U_{n+1} = u^{\alpha^{n+1}}$. A commitment to $\vec{m}$ consists of $C = g^{\gamma} \cdot \prod_{j=1}^{n} g^{\alpha^j m_j}$. Witness for a function evaluation $\langle \vec{x}, \vec{m} \rangle = y$ is defined as $W_y = \prod_{i=1}^{n} W_i^{x_i}$ with the $\mathbb{G}_{p_1}$ component of W_i being $u^{\alpha^{n-i+1}\gamma} \cdot \prod_{j=1, j \neq i}^{n} u^{\alpha^{n+1+j-i} m_j}$ for each $i = 1, \ldots, n$. The absence of U_{n+1} in the witness allows verifying that $y = \langle \vec{x}, \vec{m} \rangle$ by testing if $e(C, \prod_{i=1}^{n} u^{\alpha^{n-i+1} x_i}) = e(g^{\alpha}, u^{\alpha^n})^y \cdot e(g, W_y)$. The u-components are randomized with elements of $\mathbb{G}_{p_3}$. This modification does not affect verification since the $\mathbb{G}_{p_3}$ components get cancelled upon pairing with $\mathbb{G}_{p_1}$ elements. The scheme is a composite-order analogue of the one proposed in [22].

Proof Idea. A $(q_1 \to q_2)$ subgroup decision assumption requires random elements of $\mathbb{G}_{q_1}$ to be indistinguishable from random elements of $\mathbb{G}_{q_2}$. Using Wee's adaptation [32] of the Déjà Q framework [13], we prove that our FC scheme is computationally binding based on $(p_1 \to p_1 p_2)$ and $(p_1 p_3 \to p_1 p_2 p_3)$ subgroup decision assumptions. An adversary breaking the binding property is successful if it can produce a commitment C and two conflicting witnesses W_y and $W_{y'}$ for evaluation of a function $\vec{x}$. Given that both witnesses satisfy the verification equations, one can say that the adversary can essentially produce $\Delta W = \left(W_{y'}/W_y\right)^{1/(y-y')}$ which is of the form $u^{(\alpha^{n+1})} \cdot g_2^{r_2} \cdot g_3^{r_3}$ for some $r_2, r_3 \in \mathbb{Z}_N$ and generators $g_2 \in \mathbb{G}_{p_2}$ and

$g_3 \in \mathbb{G}_{p_3}$. The $\mathbb{G}_{p_1}$ component of ΔW is identical to that of the trapdoor key. Define two types of keys (resp. attacks) according to $\{U_j\}_{j=1}^{2n}$ (resp. ΔW) containing a $\mathbb{G}_{p_2}$ component or not. We argue that the attacker cannot mount an attack of a type different from that of the key based on the $(p_1 \to p_1 p_2)$. The distribution of $\mathbb{G}_{p_2}$ components for the keys are changed gradually via the transition described below.

$$u^{\alpha^i} R_{3,i} \xrightarrow{\text{subgroup}} u^{\alpha^i} g_2^{r_1 \alpha^i} R_{3,i} \xrightarrow{\text{CRT}} u^{\alpha^i} g_2^{r_1 \alpha_1^i} R_{3,i},$$

where α_1 is uniformly distributed over $\mathbb{Z}_N$. The first transition uses the $p_1 p_3 \to p_1 p_2 p_3$ subgroup decision assumptions and the second transition is based on the Chinese remainder theorem (CRT) that states that α mod p_1 and α mod p_2 are uncorrelated. We can thus replace α mod p_2 by α_1 mod p_2 as long as the former is unconditionally hidden from the attacker. By repeating the transition $2n$ times, we obtain the transformation: $u^{\alpha^i} \to u^{\alpha^i} g_2^{\sum_{j=1}^{2n} r_j \alpha_j^i} R'_{3,i}$.

The exponent of g_2 is a pseudorandom function [13, 32] and hence can be replaced by a random exponent, $RF(i)$ for U_i in particular. After the final transition, creating ΔW consistent with these keys amounts to predicting the value of the random function evaluated at $n+1$ (for the trapdoor U_{n+1}), which is statistically infeasible.

Polynomial Commitments from Simple Assumptions. We wish to commit to a polynomial $P[Z] = a_0 + a_1 Z + \cdots + a_{n-1} Z^{n-1}$ of degree n over $\mathcal{D}$ and reveal an opening for $P(x)$ for $x \in \mathcal{D}$. Using the FC scheme for linear functions, we can commit to $(a_0, \ldots, a_{n-1}) \in \mathcal{D}^n$ so that an opening to $P(x)$ is a witness for $\langle \vec{x}, \vec{m} \rangle = P(x)$ where $\vec{x} = (1, x, \ldots, x^{n-1})$.

Accumulators for Large Universes. An accumulator allows hashing a set to a single element so that one can prove the membership of a value in the set. Vector commitments are known to imply accumulators [10], but via a construction that only supports a small universe of values. Our polynomial commitment naturally leads to an accumulator for large universes (i.e., the domain size can be exponential in the security parameter). To accumulate a set of values $S = \{y_1, \ldots, y_{n-1}\}$, use a polynomial commitment to $P[Z] = \prod_{i=1}^{n-1}(Z - y_i)$. A witness for $x \in S$ (or $x \notin S$) is generated based on the fact $P[x] = 0$ if and only if $x \in S$.

Tackling Subset Queries. Polynomial commitments and universal accumulators can be seen as direct consequences of the FC for linear functions. On the other hand, proving security for accumulators with concise subset witnesses requires a novel extension of the Déjà Q framework. We now provide a brief outline of the same.

Let n be the maximal number of accumulated values and let d be the maximal size of "provable" subsets. In the commitment scheme, keys consisted of powers of α in the exponent over the interval $[1, 2n]$ with a hole at position $n+1$. We extend this interval to $[1, (d+1)n]$ keeping $n+1, 2n+1, \ldots, (d+1)n$ powers of α as part of the trapdoor. The witness component for a specific position i of the linear function was defined as $W_i = u^{\alpha^{n-i+1}\gamma} \cdot \prod_{j=1, j \neq i}^{n} u^{\alpha^{n+1+j-i} m_j}$. To combine witnesses for several (at most d) values into a constant-size witness, we define the witness for the i-th position of the ℓ-th element as a "shift" of W_i by n. More precisely, $W_{\ell,i}$ is defined to have $u^{\alpha^{\ell n-i+1}\gamma} \cdot \prod_{j=1, j \neq i}^{n} u^{\alpha^{\ell n+1+j-i} m_j}$ as its $\mathbb{G}_{p_1}$ component.

Security for accumulators is captured by the notion of *collision-freeness* which asserts that it is computationally infeasible for an attacker to produce a set S and a witness W_X for a subset $X = \{x_1, \ldots, x_k\} \nsubseteq S$ that verifies correctly with an accumulated value for S (generated using randomness specified by the adversary). Given the randomness, the reduction can compute valid witnesses of membership and non-membership for individual values in X (as in the normal accumulator scheme). Combining appropriate "shifts" of these witnesses gives us $W_{X \cap S}$ (combined membership witness) and $W_{X \setminus S}$ (combined non-membership witness). We then observe that $W/(W_{X \cap S} W_{X \setminus S})$ has a $\mathbb{G}_{p_1}$-component of the form $u^{\sum_{\ell \in [1,k], x_\ell \notin S} w_\ell \alpha^{\ell n+1}}$ $(w_\ell \neq 0)$ which means that the attacker essentially produces a linear combination of the discrete logarithms of trapdoor keys in the exponent. The rest of the reduction proceeds similar to the FC scheme with the pseudorandom function now extending to the larger interval. Using this pseudorandom function, the distribution of the keys is gradually modified until the $\mathbb{G}_{p_2}$ components of all U_i's are truly random. We argue that generating such a witness requires the adversary to predict a linear combination of at most d specific evaluations of a random function which is clearly infeasible.

2 Background

2.1 Bilinear Maps and Complexity Assumptions

We use groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1 p_2 p_3$ endowed with an efficiently computable map (a.k.a. pairing) $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that: (1) $e(g^a, h^b) = e(g,h)^{ab}$ for any $(g,h) \in \mathbb{G} \times \mathbb{G}$ and $a, b \in \mathbb{Z}$; (2) if $e(g,h) = 1_{\mathbb{G}_T}$ for each $h \in \mathbb{G}$, then $g = 1_{\mathbb{G}}$. Also, pairing two elements of order p_i and p_j, with $i \neq j$, always gives the identity element $1_{\mathbb{G}_T}$.

In the following, for each $i \in \{1,2,3\}$, we denote by $\mathbb{G}_{p_i}$ the subgroup of order p_i. For all distinct $i, j \in \{1,2,3\}$, we call $\mathbb{G}_{p_i p_j}$ the subgroup of order $p_i p_j$. We rely on the following assumptions introduced in [20], which are non-interactive, falsifiable. In both of them, the number of input elements is constant (regardless of the number of adversarial queries).

Assumption 1. Given a description of $(\mathbb{G}, \mathbb{G}_T)$ as well as $g \overset{R}{\leftarrow} \mathbb{G}_{p_1}, X_3 \overset{R}{\leftarrow} \mathbb{G}_{p_3}$ and $T \in \mathbb{G}$, it is infeasible to efficiently decide if $T \in \mathbb{G}_{p_1 p_2}$ or $T \in \mathbb{G}_{p_1}$.

Assumption 2. Let $g, X_1 \overset{R}{\leftarrow} \mathbb{G}_{p_1}, X_2, Y_2 \overset{R}{\leftarrow} \mathbb{G}_{p_2}, Y_3, Z_3 \overset{R}{\leftarrow} \mathbb{G}_{p_3}$. Given a description of $(\mathbb{G}, \mathbb{G}_T)$, $(g, X_1 X_2, Z_3, Y_2 Y_3)$ and T, it is hard to decide if $T \in_R \mathbb{G}_{p_1 p_3}$ or $T \in_R \mathbb{G}$.

2.2 Vector Commitment Schemes

In prime order groups, Libert and Yung [22] introduced concise vector commitment schemes, which are commitments that can be opened with a short de-commitment string for each individual coordinate. Such commitments were described in [22, 10]. In [22], the commitment key is $CK = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}) \in \mathbb{G}^{2n}$, where $g_i = g^{(\alpha^i)}$ for each i. The trapdoor is g_{n+1}. To commit to $\vec{m} = (m_1, \ldots, m_n)$, one picks $r \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $C = g^r \cdot \prod_{j=1}^n g_{n+1-j}^{m_\kappa}$. A single element $W_i = g_i^r \cdot \prod_{j=1, j \neq i}^n g_{n+1-j+i}^{m_j}$ provides evidence that m_i is the i-th component of $\vec{m}$ as it satisfies $e(g_i, C) = e(g, W_i) \cdot e(g_1, g_n)^{m_i}$. The infeasibility of opening C to two distinct messages for some i relies on a parametrized assumption [5].

2.3 Functional Commitments for Linear Functions: Definitions

In [18], Izabachène *et al.* implicitly showed that the vector commitment scheme of [22] can be generalized into a commitment scheme allowing to commit to a vector $\vec{m}$ while proving – via a partial opening made of a short piece of information – that the committed vector $\vec{m}$ satisfies $\langle \vec{m}, \vec{x} \rangle = y$, for some public $\vec{m}$ and y. We call such a primitive *functional* commitment for *linear* functions. In this section, we formally define this primitive and its security.

▶ **Definition 1** (Functional Commitments). Let $\mathcal{D}$ be a domain and consider linear functions $\langle \cdot, \cdot \rangle : \mathcal{D}^n \times \mathcal{D}^n \to \mathcal{D}$ defined by $\langle \vec{x}, \vec{m} \rangle = \sum_{i=1}^{n} x_i m_i$ for $\vec{x}, \vec{m} \in \mathcal{D}^n$ with $\vec{x} = (x_1, \dots, x_n), \vec{m} = (m_1, \dots, m_n)$. A functional commitment scheme FC for $(\mathcal{D}, n, \langle \cdot, \cdot \rangle)$ is a tuple of four (possibly probabilistic) polynomial time algorithms – (Setup,Commit,Open,Verify).

$\mathsf{Setup}(1^\lambda, 1^n)$: takes in a security parameter $\lambda \in \mathbb{N}$, a desired message length $n \in \mathsf{poly}(\lambda)$ and outputs a commitment key CK and, optionally, a trapdoor TK.

$\mathsf{Commit}(CK, \vec{m})$: takes as input the commitment key CK, a message vector $\vec{m} \in \mathcal{D}^n$ and outputs a commitment C for $\vec{m}$ and auxiliary information denotes aux.

$\mathsf{Open}(CK, C, \mathsf{aux}, \vec{x})$: takes as input the commitment key CK, a commitment C (to $\vec{m}$), auxiliary information (possibly containing $\vec{m}$) and a vector $\vec{x} \in \mathcal{D}^n$; computes a witness W_y for $y = \langle \vec{x}, \vec{m} \rangle$ *i.e.,* W_y is a witness for the fact that the linear function defined by $\vec{x}$ when evaluated on $\vec{m}$ gives y.

$\mathsf{Verify}(CK, C, W_y, \vec{x}, y)$: takes as input the commitment key CK, a commitment C, a witness W_y, a vector $\vec{x} \in \mathcal{D}^n$ and $y \in \mathcal{D}$; outputs 1 if W_y is a witness for C being a commitment for some $\vec{m} \in \mathcal{D}^n$ such that $\langle \vec{x}, \vec{y} \rangle = y$ and outputs 0 otherwise.

The correctness condition for a functional commitment scheme requires that for every $(CK, TK) \leftarrow \mathsf{Setup}(\lambda, n)$, for all $\vec{m}, \vec{x} \in \mathcal{D}^n$, if $(C, \mathsf{aux}) \leftarrow \mathsf{Commit}(CK, \vec{m})$ and $W_y \leftarrow \mathsf{Open}(CK, C, \mathsf{aux}, \vec{x})$, then $\mathsf{Verify}(CK, C, W_y, \vec{x}, y) = 1$ with probability 1.

The security requirements of functional commitments are formalized as follows. The perfect hiding property mandates that the distribution of the commitment string $\mathsf{Commit}(CK, \vec{m})$ be independent of the message $\vec{m}$.

▶ **Definition 2** (Perfectly Hiding). A commitment scheme is perfectly hiding if for a key CK generated by an honest setup, for all $\vec{m}_1, \vec{m}_2 \in \mathcal{D}^n$ with $\vec{m}_1 \neq \vec{m}_2$, the two distributions $\{CK, \mathsf{Commit}(CK, \vec{m}_1)\}$ and $\{CK, \mathsf{Commit}(CK, \vec{m}_2)\}$ are identical given that the random coins of Commit are chosen according to the uniform distribution from the respective domain.

The binding property requires the infeasibility of generating a commitment C and accepting witnesses for two distinct values y, y' without knowing the trapdoor TK.

▶ **Definition 3** (Function Binding). A functional commitment scheme $\mathsf{FC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$ for $(\mathcal{D}, n, \langle \cdot, \cdot \rangle)$ is said to be computationally binding if any PPT adversary $\mathcal{A}$ has negligible advantage in winning the following game.

1. The challenger generates (CK, TK) by running $\mathsf{Setup}(\lambda, n)$ and gives CK to $\mathcal{A}$.
2. The adversary $\mathcal{A}$ outputs a commitment C, a vector $\vec{x} \in \mathcal{D}^n$, two values $y, y' \in \mathcal{D}$ and two witnesses $W_y, W_{y'}$. We say that $\mathcal{A}$ wins the game if the following conditions hold.
   (i) $y \neq y'$;
   (ii) $\mathsf{Verify}(CK, C, W_y, \vec{x}, y) = \mathsf{Verify}(CK, C, W_{y'}, \vec{x}, y') = 1$.

2.4 Cryptographic Accumulators

The basic functionality of an accumulator is to combine a set S of values into a single value V so that for any $x \in S$ it is possible to prove that x is accumulated in V.

▶ **Definition 4** (Accumulator). Let $\mathcal{D}$ be a domain. An accumulator scheme Acc for $\mathcal{D}$ is a tuple $(\mathsf{Setup}, \mathsf{Eval}, \mathsf{WitCreate}, \mathsf{Verify})$ of PPT algorithms defined as follows.

$\mathsf{Setup}(1^\lambda, 1^n)$: takes as input a security parameter λ and an integer $n \in \mathbb{N}$ upper bounding the number of elements that can be accumulated; outputs a pair of keys (PK, SK).

$\mathsf{Eval}(PK, S)$: inputs a key PK, a set $S \subset \mathcal{D}$ of elements (with $|S| \leq n$) to be accumulated and outputs an accumulated value V along with some auxiliary information aux.

WitCreate$(PK, S, V, \mathsf{aux}, x, \mathsf{type})$: inputs a public key PK, a set S, a pair of accumulated value and state information (V, aux) generated by $\mathsf{Eval}(PK, S)$, an element $x \in \mathcal{D}$ and a boolean value $\mathsf{type} \in \{0, 1\}$ indicating whether the output should be membership or non-membership witness according as its value is 1 or 0 respectively.

Case type $= 1$: If $x \notin S$, it returns $\perp$. Otherwise, a membership witness W is returned.

Case type $= 0$: It returns $\perp$ if $x \in S$ and a non-membership witness W otherwise.

Verify$(PK, V, W, x, \mathsf{type})$: takes as input the public key PK, an accumulator V for set S, a witness W, an element $x \in \mathcal{D}$ and a boolean value type. Returns 1 if and only if either

- W is a valid witness for $x \in S$ and $\mathsf{type} = 1$
- W is a valid witness for $x \notin S$ and $\mathsf{type} = 0$.

The above definition consider static accumulators. In dynamic accumulators, the accumulated value as well as witnesses can be publicly updated whenever an element is added to or deleted from the set. In this work, we only consider static accumulators.

The correctness condition requires that for all honestly generated keys, all honestly computed accumulators and witnesses, the Verify algorithm always accepts. An accumulator scheme is deemed secure if it is at least *collision-free*. Collision-freeness ensures the computational infeasibility of producing either a membership witness for a non-accumulated value or a non-membership witness for an accumulated value.

In accumulators supporting subset queries, witnesses can be generated for a subset of the accumulated set rather than individual elements. While accumulators have been defined in the universal setting, i.e., both membership and non-membership witnesses can be generated, here we only consider the non-universal setting.

▶ **Definition 5** (Accumulator with subset queries)**.** Let $\mathcal{D}$ be a domain. An accumulator scheme Acc for $\mathcal{D}$ is defined by a tuple (Setup, Eval, WitCreate, Verify) of probabilistic polynomial time algorithms defined as follows.

Setup$(1^\lambda, 1^n, 1^d)$: takes as input a security parameter λ, an upper bound $n \in \mathbb{N}$ on the number of elements that can be accumulated and an integer $d \in \mathbb{N}$ denoting the maximum size of a set for which a witness can be created; outputs a pair of keys (PK, SK).

Eval(PK, S): takes in a public key PK, a set $S \subset \mathcal{D}$ of elements (with $|S| \leq n$) to be accumulated and outputs an accumulated value V with some auxiliary information aux.

WitCreate$(PK, S, V, \mathsf{aux}, X)$: inputs a public key PK, a set S, a pair of accumulated value and state information (V, aux) generated by $\mathsf{Eval}(PK, S)$, a set $X \subseteq S$ with $|X| \leq d$ and outputs a witness W_X.

Verify(PK, V, W_X, X): takes as input the public key PK, an accumulator V for set S, a witness W_X, a set $X \subseteq S$. Returns 1 if W_X is a witness for $X \subseteq S$ and $\perp$ otherwise.

In the above syntax, we assume that the auxiliary information aux includes the randomness that was used to compute V when Eval is a probabilistic algorithm.

3 A Functional Commitment from Subgroup Decision Assumptions

Here, we prove that the Déjà Q framework [13] allows proving the security of the functional commitment of [18] under constant size assumptions by switching to composite order groups.

Setup$(1^\lambda, 1^n)$: Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1p_2p_3$, where $p_i > 2^{l(\lambda)}$ for each $i \in \{1, 2, 3\}$, for a suitable polynomial $l : \mathbb{N} \to \mathbb{N}$. Choose $g, u \overset{R}{\leftarrow} \mathbb{G}_{p1}$, $R_3 \overset{R}{\leftarrow} \mathbb{G}_{p_3}$ and $\alpha \overset{R}{\leftarrow} \mathbb{Z}_N$ at random in order to define $G_j = g^{\alpha^j}$ for each $j \in [1, n]$ and

$$\begin{array}{rclcl} U_1 & = & u^{\alpha} \cdot R_{3,1}, & \dots & U_n = u^{(\alpha^n)} \cdot R_{3,n}, \\ U_{n+2} & = & u^{(\alpha^{n+2})} \cdot R_{3,n+2}, & \dots & U_{2n} = u^{(\alpha^{2n})} \cdot R_{3,2n}, \end{array}$$

where $R_{3,j} \stackrel{R}{\leftarrow} \mathbb{G}_{p_3}$ for each $j \in [1,2n]\backslash\{n+1\}$. Define the commitment key to consist of $CK := \big(g,\ \{G_j\}_{j=1}^{n},\ \{U_j\}_{j\in[1,2n]\backslash\{n+1\}},\ R_3\big)$. The trapdoor consists of $TK := U_{n+1} = u^{(\alpha^{n+1})} \cdot R_{3,n+1}$, where $R_{3,n+1} \stackrel{R}{\leftarrow} \mathbb{G}_{p_3}$.

Commit($CK, \vec{m}$): Given $\vec{m} = (m_1,\dots,m_n) \in \mathbb{Z}_N^n$, compute $C = g^{\gamma} \cdot \prod_{j=1}^{n} G_j^{m_j}$ for a random choice of $\gamma \stackrel{R}{\leftarrow} \mathbb{Z}_N$ and output C with the auxiliary information $\mathsf{aux} = (m_1,\dots,m_n,\gamma)$.

Open($CK, C, \mathsf{aux}, \vec{x}$): Given $\vec{x} = (x_1,\dots,x_n) \in \mathbb{Z}_N^n$, the auxiliary information $\mathsf{aux} = (m_1,\dots,m_n,\gamma)$ allows generating a witness for the function $\langle \vec{m},\vec{x}\rangle = \sum_{i=1}^{n} m_i \cdot x_i$ by computing

$$W_i = U_{n-i+1}^{\gamma} \cdot \prod_{j=1,j\neq i}^{n} U_{n+1+j-i}^{m_j} \qquad \forall i \in \{1,\dots,n\}, \tag{1}$$

and outputting $W_y = \prod_{i=1}^{n} W_i^{x_i}$.

Verify($CK, C, W_y, \vec{x}, y$): Given $C \in \mathbb{G}$ and $\vec{x} = (x_1,\dots,x_n) \in \mathbb{Z}_N^n$, accept $W_y \in \mathbb{G}$ as evidence that C is a commitment to $\vec{m} \in \mathbb{Z}_N^n$ such that $y = \langle \vec{m},\vec{x}\rangle$ if and only if it holds that $e(C, \prod_{i=1}^{n} U_{n-i+1}^{x_i}) = e(G_1,U_n)^{y} \cdot e(g,W_y)$. If so, output 1. Otherwise, return 0.

The correctness is verified by observing that, for each $i \in \{1,\dots,n\}$, (1) implies that

$$e(C,U_{n-i+1}) = e(g,u)^{(\alpha^{n+1})m_i} \cdot e\big(g, U_{n-i+1}^{\gamma} \prod_{j=1,j\neq i}^{n} U_{n+j-i+1}^{m_j}\big) = e(G_1,U_n)^{m_i} \cdot e(g,W_i).$$

By raising both members of the above equality to the power $x_i \in \mathbb{Z}_N$ and taking the product over all $i \in [1,n]$, we find that W_y satisfies $e(C, \prod_{i=1}^{n} U_{n-i+1}^{x_i}) = e(G_1,U_n)^{\langle \vec{m},\vec{x}\rangle} \cdot e(g,W_y)$.

It is clear that that the commitment is perfectly hiding: since C lives in the cyclic subgroup $\mathbb{G}_{p_1}$, any vector $(m_1,\dots,m_n) \in \mathbb{Z}_N^n$ has a corresponding opening $\gamma \in \mathbb{Z}_N$ (and even p_2p_3 openings since only $\gamma \bmod p_1$ is fixed by $\vec{m}$).

We now prove it computationally binding under subgroup assumptions. While this property can be proved via a reduction from the one-wayness of Wee's broadcast encryption [32, Section 4], we found it interesting to give a direct proof under the underlying assumptions for two reasons. First, this proof allows relying on a computational (rather than decisional) analogue of Assumption 1. Second, the proof provides insights allowing to prove the security of variants of this commitment or the other primitives it implies. For example, by adapting the proof of Theorem 6, we design an accumulator supporting subset queries in section 5. Since the latter scheme has a public key containing more elements than in [32], it can hardly be proved secure via a reduction from the security of Wee's broadcast encryption [32].

The proof involves two computationally indistinguishable distributions of parameters (CK, TK). The normal distribution is as in the real scheme whereas the semi-functional distribution allows CK and TK to have a $\mathbb{G}_{p_2}$ component. As in [32, Theorem 2], we use the Déjà Q framework so as to gradually move to a game where the $\{U_i\}_{i=1}^{2n}$ all contain a $\mathbb{G}_{p_2}$ component $g_2^{R(i)}$ which is determined by a random function $R : [1,2n] \to \mathbb{Z}_{p_2}$. As in [22, 18], we rely on the fact that any attack against the binding property publicly reveals a value U_{n+1} which contains $u^{(\alpha^{n+1})}$ as its $\mathbb{G}_{p_1}$ component. Depending on whether U_{n+1} contains a $\mathbb{G}_{p_2}$ component or not, we speak of Type B or Type A attacks. The proof uses a subsequence of $2n$ games where, in the k-th game, the $\mathbb{G}_{p_2}$ component of U_i is of the form $g_2^{F_k(i)}$, where $F_k : [1,2n] \to \mathbb{Z}_{p_2}$ is a k-wise independent function. The strategy of the proof is to show that, unless either Assumption 1 or Assumption 2 can broken, the attack on the binding property also reveals a U_{n+1} of the form $U_{n+1} = u^{(\alpha^{n+1})} \cdot g_2^{F_k(n+1)} \cdot \mathcal{R}_3$, for some $R_3 \in \mathbb{G}_{p_3}$ in the k-th game. Said otherwise, the attack reveals a trapdoor U_{n+1} which mimics the distribution of the commitment key CK. When we reach the $2n$-th game, the $\mathbb{G}_{p_2}$ component of each U_i is determined by $F_{2n}(i)$. Since $F_{2n}(.)$ is a $2n$-wise independent function, the $\mathbb{G}_{p_2}$ of U_{n+1}

is thus statistically independent of those of $\{U_i\}_{i\in[1,2n]\setminus\{i\}}$, which appear in the public key. The proof of Theorem 6 is given in the full version of the paper.

▶ **Theorem 6.** *The scheme is binding if Assumption 1 and Assumption 2 both hold.*

4 Further Constructions

4.1 Polynomial Commitments from Constant-Size Assumptions

It is easy to see that any functional commitment for linear functions implies a polynomial commitment. Indeed, in order to commit to a polynomial $P[Z] = a_0+a_1Z+\cdots+a_{n-1}Z^{n-1}$ of degree $n-1$, we can simply commit to the vector of coefficients $\vec{m} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{Z}_N^n$. When the sender wants to convince a verifier that $P(x) = y$, for some public $x, y \in \mathbb{Z}_N$, it is sufficient to generate a witness W_y showing that $\langle\vec{m}, \vec{x}\rangle = y$, where $\vec{x} = (1, x, x^2, \ldots, x^{n-1})$. Our construction of Section 3 thus implies the first polynomial commitment based on constant-size assumptions. Indeed, the schemes of [19, 7] rely on q-type assumptions where q is proportional to the maximal degree of committed polynomials.

4.2 Large-Universe Pairing-Based (Universal) Accumulators from Constant-Size Assumptions

Catalano and Fiore [10] designed cryptographic accumulators from vector commitments. While their construction yields an accumulator based on the Diffie-Hellman assumption, it only supports small universes. Namely, accumulated values should come from a polynomial-size domain since the public key has linear size in the cardinality of this domain.

It is easy to see that polynomial commitments imply accumulators for exponential-size universes. While the size of the public key is linear in the maximal number of accumulated values (as in Nguyen's accumulator [26]), it does not depend of the universe size. As a result, we can accumulate inputs consisting of arbitrary strings of polynomial length.

In order to accumulate a set $S = \{x_1, \ldots, x_{n-1}\}$, one can commit to the vector $(a_0, a_1, \ldots, a_{n-2}, 1)$ that contains the coefficients of the polynomial $P[Z] = \prod_{j=1}^{n-1}(Z - x_j)$ and rely on the fact that $x \in S$ if and only if $P(x) = 0$. A witness that $x_i \in S$ (resp. $x_i \notin S$) is obtained by generating a witness that the committed polynomial satisfies $P(x_i) = 0$ (resp. $P(x_i) \neq 0$)). A concrete construction based on Assumptions 1 and 2 is described in the the full version.

5 Accumulators Supporting Subset Queries

We now generalize the accumulator of Section 4.2 so that a constant-size witness $W \in \mathbb{G}$ can provide evidence that a purported set X is contained in the hashed set S. Such a commitment was previously designed by Papamanthou *et al.* [29] under a non-standard q-type assumption. Our construction is thus the first realization based on fixed-size assumptions.

Gen$(1^\lambda, 1^n)$: Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1p_2p_3$, where $p_i > 2^{l(\lambda)}$ for each $i \in \{1, 2, 3\}$, for a suitable polynomial $l : \mathbb{N} \to \mathbb{N}$. Choose $g, u \xleftarrow{R} \mathbb{G}_{p_1}$, $R_3 \xleftarrow{R} \mathbb{G}_{p_3}$ and $\alpha \xleftarrow{R} \mathbb{Z}_N$ at random. Let $d \leq n$ be the bound placed on size of a subset (also polynomial in the security parameter). Define $G_i = g^{(\alpha^i)}$ for each $i \in [1, n]$ and $U_j = u^{(\alpha^j)} \cdot R_{3,j}$, where $R_{3,j} \xleftarrow{R} \mathbb{G}_{p_3}$ for each $j \in [1, (d+1)n] \setminus \{n+1, 2n+1, \ldots, dn+1\}$.

The secret key is $SK := \{U_{\ell n+1}\}_{\ell=1}^{d}$, where $U_{\ell n+1} = u^{(\alpha^{\ell n+1})} \cdot R_{3,\ell n+1}$ with $R_{3,\ell n+1} \overset{R}{\leftarrow} \mathbb{G}_{p_3}$ for all $\ell \in [1, d]$. The public key is

$$PK := \big(g, \{G_j\}_{j=1}^{n}, \{U_j\}_{j\in[1,(d+1)n]\setminus\{n+1,2n+1,\ldots,dn+1\}}, R_3\big).$$

Eval(PK, S): To hash a set $S = \{y_1, \ldots, y_{n'}\}$ of cardinality $n' \le n-1$, expand the polynomial $P_S[Z] = \prod_{j=1}^{n'}(Z - y_j) = \sum_{j=0}^{n'} m_j \cdot Z^j$. Choose $\gamma \overset{R}{\leftarrow} \mathbb{Z}_N$ to compute and output

$$V = g^{\gamma} \cdot \prod_{j=1}^{n'+1} G_j^{m_{j-1}} = g^{\gamma+\alpha\cdot P_S(\alpha)}, \qquad \mathsf{aux} = (S, \gamma) \tag{2}$$

WitCreate($PK, V, S, \mathsf{aux}, X$): Given a set $S = \{y_1, \ldots, y_{n'}\}$, a subset $X = \{x_1, \ldots, x_k\} \subseteq S$ of size $k \le d$ (we assume w.l.o.g. that $x_1, \ldots, x_k$ are arranged in some pre-determined lexicographical order), and the state information $\mathsf{aux} = (S, \gamma)$ such that (V, aux) was produced by $\mathsf{Acc}(PK, S)$, compute $P_S[Z] = \prod_{j=1}^{n'}(Z - y_j) = \sum_{j=0}^{n'} m_j \cdot Z^j$ and define the corresponding vector $\vec{m} = (m_0, m_1, \ldots, m_{n'}, 0, \ldots, 0) \in \mathbb{Z}_N^n$. For each $\ell \in [1, k]$, define $\vec{x}_\ell = (x_{\ell,1}, \ldots, x_{\ell,n}) = (1, x_\ell, x_\ell^2, \ldots, x_\ell^n) \in \mathbb{Z}_N^n$ which satisfies $P_S(x_\ell) = \vec{m} \cdot \vec{x}_\ell = 0$. For $\ell \in [1, k]$, generate a witness that $\langle \vec{m}, \vec{x}_\ell \rangle = 0$ by first using $\{U_{\ell n+1+j-i}\}_{j \ne i}$ to compute

$$W_{\ell,i} = U_{\ell n-i+1}^{\gamma} \cdot \prod_{j=1, j\ne i}^{n} U_{\ell n+1+j-i}^{m_j} \qquad \forall i \in \{1, \ldots, n\}, \tag{3}$$

which satisfies $e(V, \prod_{i=1}^{n} U_{\ell n+1-i}^{x_{\ell,i}}) = e(g, W_{\ell,i})$ for all $\ell \in [1, k]$ since $\vec{m} \cdot \vec{x}_\ell = 0$. Then, compute and output the witness $W_X = \prod_{\ell=1}^{k} \prod_{i=1}^{n} W_{\ell,i}^{x_{\ell,i}}$.

Verify(PK, V, W_X, X): Given an accumulator value $V \in \mathbb{G}$, a subset $X = \{x_1, \ldots, x_k\}$, where $x_i \in \mathbb{Z}_N$ for each $i \in [1, k]$, and a candidate a witness W_X, do the following.

1. For each $\ell \in [1, k]$, define $\vec{x}_\ell = (x_{\ell,1}, \ldots, x_{\ell,n}) = (1, x_\ell, \ldots, x_\ell^n) \in \mathbb{Z}_N^n$.
2. Return 1 if and only if $e(V, \prod_{\ell=1}^{k} \prod_{i=1}^{n} U_{\ell n+1-i}^{x_{\ell,i}}) = e(g, W_X)$.

From an efficiency standpoint, the size of PK is quadratic in n when $d \approx n$ so as to handle queries for arbitrary subsets of size $\le n$. In comparison with [29], we thus achieve security under simple assumptions at the expense of a somewhat larger public key. We see it as an interesting open problem to retain $O(n)$-size public keys under simple assumptions.

We prove that the scheme provides collision-freeness (a detailed definition is given in the full version) in that no PPT adversary can output a set S (of size $\le n$) along with a verifying witness W_X for another set X which is *not* contained in S. We thus use a natural analogue of the definition of collision-freeness used in [15]: since our evaluation algorithm is randomized, we assume that the adversary outputs the set S and the random coins γ of the evaluation algorithm that lead to the accumulator value for which W_X verifies.

The proof relies on the fact that the adversary outputs both the hashed set S and the random coins γ of the hashing algorithm. It allows the reduction to use W_X in order to extract a membership witness for the difference $X \setminus S$ using the homomorphic properties of the underlying commitment. Having obtained $W_{X\setminus S}$, the reduction is also able to compute an aggregation of non-membership witnesses for the same difference $X \setminus S$. From these two conflicting witnesses, it is possible to extract some linear combination of the secret key components $\{U_{\ell n+1}\}_{\ell=1}^{d}$. In turn, this forces the adversary to predict a linear combination of random function evaluations in the final step of the sequence of games. The proof is given in the full version of the paper.

▶ **Theorem 7.** *The scheme is collision-free if Assumption 1 and Assumption 2 hold.*

References

1 Niko Baric and Birgit Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees. In *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 480–494. Springer, 1997.

2 Josh Cohen Benaloh and Michael de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Sinatures (Extended Abstract). In *EUROCRYPT'93*, volume 765 of *LNCS*, pages 274–285. Springer, 1993.

3 Manuel Blum, Paul Feldman, and Silvio Micali. Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract). In *STOC 1988*, pages 103–112. ACM, 1988.

4 Dan Boneh and Henry Corrigan-Gibbs. Bivariate Polynomials Modulo Composites and Their Applications. In *ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 42–62. Springer, 2014.

5 Dan Boneh, Craig Gentry, and Brent Waters. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, 2005.

6 Dan Boneh, Amit Sahai, and Brent Waters. Functional Encryption: Definitions and Challenges. In *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, 2011.

7 Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev, and Markulf Kohlweiss. Composable and Modular Anonymous Credentials: Definitions and Practical Constructions. In *ASIACRYPT 2015*, volume 9453 of *LNCS*, pages 262–288. Springer, 2015.

8 Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *PKC 2009*, volume 5443 of *LNCS*, pages 481–500. Springer, 2009.

9 Jan Camenisch and Anna Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.

10 Dario Catalano and Dario Fiore. Vector Commitments and Their Applications. In *PKC 2013*, volume 7778 of *LNCS*, pages 55–72. Springer, 2013.

11 Dario Catalano, Dario Fiore, and Mariagrazia Messina. Zero-Knowledge Sets with Short Proofs. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 433–450. Springer, 2008.

12 Melissa Chase, Alexander Healy, Anna Lysyanskaya, Tal Malkin, and Leonid Reyzin. Mercurial Commitments with Applications to Zero-Knowledge Sets. In *EUROCRYPT 2005, Proceedings*, volume 3494 of *LNCS*, pages 422–439. Springer, 2005.

13 Melissa Chase and Sarah Meiklejohn. Déjà Q: Using Dual Systems to Revisit q-Type Assumptions. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 622–639. Springer, 2014.

14 Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract). In *FOCS 1985*, pages 383–395. IEEE Computer Society, 1985.

15 David Derler, Christian Hanser, and Daniel Slamanig. Revisiting Cryptographic Accumulators, Additional Properties and Relations to Other Primitives. In *CT-RSA 2015*, volume 9048 of *LNCS*, pages 127–144. Springer, 2015.

16 Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic Functions. *J. ACM*, 50(6):852–921, 2003.

17 Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled Fully Homomorphic Signatures from Standard Lattices. In *STOC 2015*, pages 469–477. ACM, 2015.

18 Malika Izabachène, Benoît Libert, and Damien Vergnaud. Block-Wise P-Signatures and Non-interactive Anonymous Credentials with Efficient Attributes. In *IMACC 2011*, volume 7089 of *LNCS*, pages 431–450. Springer, 2011.

19 Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, 2010.

20 Allison B. Lewko and Brent Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010, Proceedings*, volume 5978 of *LNCS*, pages 455–479. Springer, 2010.

21 Jiangtao Li, Ninghui Li, and Rui Xue. Universal Accumulators with Efficient Nonmembership Proofs. In *ACNS 2007*, volume 4521 of *LNCS*, pages 253–269. Springer, 2007.

22 Benoît Libert and Moti Yung. Concise Mercurial Vector Commitments and Independent Zero-Knowledge Sets with Short Proofs. In *TCC 2010, Proceedings*, volume 5978 of *LNCS*, pages 499–517. Springer, 2010.

23 Helger Lipmaa. Secure Accumulators from Euclidean Rings without Trusted Setup. In *ACNS 2012*, volume 7341 of *LNCS*, pages 224–240. Springer, 2012.

24 Silvio Micali, Michael O. Rabin, and Joe Kilian. Zero-Knowledge Sets. In *FOCS 2003*, pages 80–91. IEEE Computer Society, 2003.

25 Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable Random Functions. In *FOCS'99*, pages 120–130. IEEE Computer Society, 1999.

26 Lan Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 275–292. Springer, 2005.

27 Rafail Ostrovsky, Charles Rackoff, and Adam D. Smith. Efficient Consistency Proofs for Generalized Queries on a Committed Database. In *ICALP 2004*, volume 3142 of *LNCS*, pages 1041–1053. Springer, 2004.

28 Charalampos Papamanthou, Elaine Shi, Roberto Tamassia, and Ke Yi. Streaming Authenticated Data Structures. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 353–370. Springer, 2013.

29 Charalampos Papamanthou, Roberto Tamassia, and Nikos Triandopoulos. Optimal Verification of Operations on Dynamic Sets. In *CRYPTO 2011*, volume 6841 of *LNCS*, pages 91–110. Springer, 2011.

30 Torben P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, 1991.

31 Amit Sahai and Brent Waters. Fuzzy Identity-Based Encryption. In *EUROCRYPT 2005, Proceedings*, volume 3494 of *LNCS*, pages 457–473. Springer, 2005.

32 Hoeteck Wee. Déjà Q: Encore! Un Petit IBE. In *TCC 2016-A*, volume 9563 of *LNCS*, pages 237–258. Springer, 2016.